NAVAL SHIPBOARD NON-TACTICAL ADP PROGRAM (SNAP)

AUTOMATED MEDICAL SYSTEM (SAMS)

COMPUTER ASSISTED MEDICAL DIAGNOSIS (CAMD) MODULE

SYSTEM/SUBSYSTEM SPECIFICATION

T. R. Bonifield

J. D. Felson

Report No. 92-27

NAVAL HEALTH RESEARCH CENTER
P.O. BOX 85122
SAN DIEGO, CALIFORNIA 92186-5122

NAVAL MEDICAL RESEARCH AND DEVELOPMENT COMMAND
BETHESDA, MARYLAND

NAVAL
SHIPBOARD NON-TACTICAL ADP PROGRAM (SNAP)
AUTOMATED MEDICAL SYSTEM (SAMS)
COMPUTER ASSISTED MEDICAL DIAGNOSIS (CAMD) MODULE
SYSTEM/SUBSYSTEM SPECIFICATION

Working Draft
Version 1.1
30 September 1992

Prepared for
Naval Medical Research and Development Command
Naval Health Research Center

Prepared by
Public Health Service
Operational Medicine Informatics Laboratory

Report No. 92-27, supported by Naval Medical Research and Development Command, Bethesda, Maryland, Department of the Navy. The views expressed in this article are those of the authors and do not reflect the official policy or position of the Department of the Navy, Department of Defense, or the U.S. Government. Approved for public release; distribution unlimited.

PREFACE

This document has been prepared on the basis of an inter-service Memorandum of Agreement (MOA) between the Naval Medical Research and Development Command (NMRDC), Naval Health Research Center (NHRC) in San Diego, California and the United States Public Health Service Operational Medicine Informatics Laboratory in Monterey, California. The Shipboard Non-tactical ADP Program (SNAP) Automated Medical System (SAMS) Computer Assisted Medical Diagnosis (CAMD) Module System/Subsystem Specification (SS) was prepared according to DOD-STD-7935A, DOD Automated Information Systems (AIS) Documentation Standards, dated 31 October 1988. In particular, the CAMD SS meets the requirements for system/subsystem specifications that are defined in Section 5.2 of DOD-STD-7935A.

Thomas R. Bonifield, Ph.D.
Chief Scientist
United States Public Health Service

James D. Felson, M.D.
Medical Director
United States Public Health Service

TABLE OF CONTENTS

LIST OF APPENDIXES

LIST OF FIGURES

LIST OF TABLES

SECTION 1. GENERAL

1.1 Purpose of the System/Subsystem Specification. The System/Subsystem Specification (SS) for the Naval Shipboard Non-tactical ADP Program (SNAP) Automated Medical System (SAMS) Computer Assisted Medical Diagnosis (CAMD) Module is written to fulfill the following objectives:

a. To provide a detailed definition of the system/subsystem functions.

b. To communicate details of the ongoing analysis between the user's operational personnel and the appropriate development personnel.

c. To define in detail the interfaces with other systems and subsystems and the facilities to be utilized for accomplishing the interfaces.

1.2 Project References. Computer Assisted Medical Diagnosis (CAMD) will provide an automated medical consultation module that will help Independent Duty Corpsmen in isolated operational settings without physicians, such as small surface ships, submarines, or isolated bases, arrive at a diagnosis when presented with clinical problems. The CAMD module will also provide therapeutic plan formulation support for specific diagnoses or disorders. The CAMD module is being developed as a medical research effort. The CAMD project is focused on the medical module that can be used for the Shipboard Non-tactical ADP Programs (SNAP) Automated Medical System (SAMS) to fulfill the medical capability requirement of the SNAP Mission Element Needs Statement (MENS).

The CAMD module has three basic components: Diagnostic Application Software, a User Interface Subsystem, and a Data Directory Subsystem. The Data Directory Subsystem provides core utilities used by the User Interface Subsystem and by application programmers and diagnostic algorithm developers to implement, maintain, and augment the data files within the FoxPro software environment.

The CAMD research and development has been funded by the Naval Medical Research and Development Command (NMRDC), Bethesda, Maryland. The CAMD module will be developed in parallel with SAMS operations. The SAMS system will continue operations at all of the sites where it is deployed and will not be delayed in its own ongoing enhancement by the CAMD module not being immediately available. As CAMD features are implemented, tested, and accepted for integration into SAMS, they will be added to the SAMS system with no programming required to install them. SAMS currently is in use at over 500 Navy facilities.

The following references are applicable to the history and development of the CAMD project:

Documented Navy Requirements:

a. CNO ltr ser 987/5239822 of 9 July 1979: Science and Technology Objectives.

b. Chief BUMED ltr ser 10408029 of 15 Apr 1981: Evaluation at sea of a computer-based system to aid medical decision-making aboard submarines.

c. COMSUBLANT ltr FF4-12:004, ser 2309 of 23 May 1981 to CNO (OP-983): Evaluation at sea of a computer-based system to aid medical decision-making aboard submarines; request for Fleet T&E support of.

d. NMRDC ltr ser 3910 to CNO (OP-983) of 9 Dec 1981: Fleet T&E support for computer-assisted medical diagnosis system, K863 endorsed Chief BUMED ser 11209020.

e. Tentative Operational Requirement (TOR) developed in the Topical: Review: Combat Medical Material Research and Development OUSDRE, during 20-21 Nov 1984 at the Naval Submarine Medical Research Laboratory (Submarine Computer-Based Patient Management Systems).

f. Tri-Service Medical Information System, Functional Description for Shipboard Non-tactical ADP Program (SNAP) Automated Medical System (SAMS), March 1986.

g. Medical Requirement (MR) No. 3d of 14 December 1987: Computer-Aided Diagnosis for Submarine Systems.

h. CNO ltr 3900 ser 093 (933D4)/007 of 5 Jan 1988: Biomedical Research, Development, Test and Evaluation (RDT&E) Requirements Review Board: Minutes of 18 Dec Meeting.

i. Melaragno, NMRDC ltr 3900 ser 04/000584 of 3 May 1991: Restruct Prog Mngmt CAD Program Give NHRC Medical Decisions Dept. oversight of program.

j. Naval Shipboard Non-tactical ADP Program (SNAP) Automated Medical System (SAMS) Computer Assisted Medical Diagnosis (CAMD) Module Functional Description, Working Draft Version 3.1, 30 May 1992.

k. DD 1498, 63706N - M0095.005-6103, Medical management tools.

1. DD 1498, 63706N - M0095.005-5010, Submarine deployable computer based system for enhanced medical practice, performance, and quality.

Relevant Navy Technical References:

a. Caras, B.G., Southerland, D.G., & Fisherkeller, K.D. MEDIC - ABDOMINAL PAIN: A Decision Support Program for the Management of Acute Abdominal Pain - USER'S MANUAL (NSMRL Report 1146). Groton, CT: Naval Submarine Medical Research Laboratory, 1989.

b. Dunbar, J., & Gino, A. Neural Networks and Their Possible Use in Computer-Assisted Diagnosis (Report No. 89-42). San Diego, CA: Naval Health Research Center, 1989.

c. Pugh, W.M., & Ryman, D.H. Comparisons of Regression and Neural Network Solutions to Known Functions (Report No. 91-33). San Diego, CA: Naval Health Research Center, 1991.

d. Ryman, D.H. Computer Assisted Medical Diagnosis Problems, and Methods To Minimize Their Effects (Report No. 91-32). San Diego, CA: Naval Health Research Center, 1991.

Other Relevant Technical Sources:

a. Barnett, G.O. The Computer and Clinical Judgment, The New England Journal of Medicine, 307 (1982), 493-494.

b. Barnett, G.O., Cimino, J.J., Hupp, J.A., & Hoffer, E.P. DXplain: An Evolving Diagnostic Decision-Support System, JAMA, 258 (1987), 67-74.

c. Blois, M.S. Clinical Judgment and Computers, The New England Journal of Medicine, 303 (1980), 192-197.

d. Charniak, E. The Bayesian Basis of Common Sense Medical Diagnosis, Proceedings of the National Conference on AI (pp. 70-73). AAAI, 1983.

e. Davis, A.M. Software Requirements: Analysis and Specification. New York: Prentice-Hall, Inc., 1990.

f. Dobbins, R.W. Computer Assisted Medical Diagnostic System for Medical Practice Support System, The Johns Hopkins University, Applied Physics Laboratory, 5 Dec 1990 (Revision 1.5).

g. Dobbins, R.W. Entity-Relationship Diagrams for Medical Practice Support System, The Johns Hopkins University, Applied Physics Laboratory, 7 Jan 1991 (Revision 1.2).

h. Eberhardt, R.C., & Dobbins, R.W. Neural Network Versus Bayesian Diagnosis of Appendicitis, Proceedings of the Eleventh Annual IEEE EMBS Conference (pp. 78-80). Philadelphia, PA: November 1990.

i. Eberhardt, R.C., & Dobbins, R.W. (Eds). Neural Network PC Tools: A Practical Guide. San Diego, CA: Academic Press, 1990.

j. Gino, A., Pugh, W.M., & Ryman, D.H. MUMPS Based Integration of Disparate Computer-Assisted Medical Diagnosis Modules, MUG Quarterly, 20 (1990), 68-74.

k. Kazic, T., Lusk, E., Olson, R., Overbeek, R., & Tuecke, S. Prototyping Databases in Prolog. In L. Sterling (Ed.), The Practice of Prolog. Boston: MIT Press, 1990.

l. Ledley, R.S., & Lusted, L.B. Reasoning Foundation of Medical Diagnosis: Symbolic Logic, Probability, and Value Theory Aid in Understanding of How Physicians Reason, Science, 130 (1959), 9-22.

m. Miller, R.A., Masarie, F.E., & Myers, J.D. Quick Medical Reference (QMR) for Diagnostic Assistance, M.D. Computing, 3 (1986), 34-48.

n. Miller, R.A., Pople, H.E., & Myers, J.D. INTERNIST-1, An Experimental Computer-Based Diagnostic Consultant for General Internal Medicine, The New England Journal of Medicine, 307 (1982), 468-476.

o. Reggia, J.A., & Perricone, B.T. Answer Justification in Medical Decision Support Systems Based on Bayesian Classification, Comput. Biol. Med., 15:4 (1985), 161-167.

p. Robinson, K.D., Ryack, B.L., Moeller, G., Post, R., & Shroeder, R.W. A Computer-Based Diagnostic/Patient Management System for Isolated Environments, Methods of Information in Medicine, 22 (1983), 131-134.

q. Shlaer, S., & Mellor, S.J. Object-oriented Systems Analysis. New York: Yourdon Press, 1988.

r. Shortliffe, E.H. Computer-Based Medical Consultations: MYCIN. New York: Elsevier/North Holland, 1976.

s. Stetson, D.M., Eberhardt, R.C., Dobbins, R.W., Pugh, W.M., & Gino, A. Structured Specification of a Computer Assisted Medical Diagnostic System, Third Annual IEEE Computer-Based Medical Systems Symposium, Chapel Hill, North Carolina, June 1990.

t. Weiss, S.M., Kulikowski, C.A., Amarel, S., & Safir, A. A Model-Based Method for Computer-Aided Medical Decision Making, Artif. Intel., 11 (1978), 145-172.

Applicable Standards Documentation:

a. Federal Information Processing Standards Publication 41, Computer Security Guidelines for Implementing the Privacy Act of 1974, 30 May 1975.

b. NAVDAC Publication 24.1, Life Cycle Management - Navy Data Automation Management Practices and Procedures: Project Management, 9 Mar 1983.

c. NAVDAC Publication 24.2, Life Cycle Management - Navy Data Automtation Management Practices and Procedures: System Decisions, 9 Mar 1983.

d. DOD-STD-7935A, DOD Automated Information Systems (AIS) Documentation Standards, 31 Oct 1988.

1.3 Terms and Abbreviations.

1.3.1 Terms. The following terms are used in this System/ Subsystem Specification (SS).

Applications: Computer software programs, written in any programming language, that meet the needs of a particular problem or perform a specified function (e.g., CAMIS, Lotus 1-2-3, NOHIMS).

Artificial Intelligence (AI): A field of computer science that deals with problem solving by computer systems modeled after human intelligence.

Artificial Neural Networks: Massively parallel computing paradigms that involve many simple processing elements used to derive output values from a set of inputs. Artificial Neural Networks were developed to emulate neuronal structure and function of the brain.

Bayesian Method: A method for computing the posterior probability of an event from information on the prior probability of the event and associated conditional probabilities using Bayes Theorem: $P(A|B) = [P(A)*P(B|A)]/P(B)$.

Bindings: The set of rules and constraints that facilitate the linkage between two operations. In Figure 1 for example, between the Application Topic Function component (2) and the Topic Knowledge Controller Utility (3), the bindings would demand of the Application Topic Function (2) the identity of the topic

function and the subject that was selected to be passed as a parameter in a predetermined format.

Conditional Probability: The probability of each possible attribute given an event has occurred. For example, each possible combination of signs and symptoms given that a specific disease has occurred.

Control Attributes: Initially control attributes define the knowledge element data type. For a particular type, it is then the collection of attributes and constraints that govern the entry, manipulation (handling), use, and external presentation (output) of a particular knowledge element.

Database Management (DBM): A management approach to database design consisting of deciding what has to be collected, stored, and processed by computer software.

Domain: A specific subject area such as diagnoses associated with chest pain. Expert systems are best used when they are applied to a very specific, circumscribed domain.

Expert Rule-Based Systems: Artificial Intelligence (AI) computer systems that consist of knowledge bases and inference engines that typically rely on IF-THEN-ELSE type rule processing to suggest answers.

Expert System: A computer program that embodies knowledge of a particular domain in conjunction with inferencing mechanisms that enable these mechanisms to use this knowledge to advise, analyze, categorize, communicate, consult, design, diagnose, explain, explore, forecast, form concepts, identify, interpret, justify, learn, manage, monitor, plan, present, retrieve, schedule, test, and tutor.

Expert System Shell: A domain-independent software tool for building expert systems. Minimally, a shell includes an inference engine, some means of entering rules or examples, and some means of using the system in a consultation mode. Shells may also provide a variety of tools that can be used in maintaining and debugging the knowledge base, and for interfacing with external programs and data. An expert system shell provides an empty framework into which users must add rules and facts.

Hybrid System: A system that implements a combination of computational methods for arriving at a result. For example, Bayesian and neural network procedures may be combined to reach a suggested medical diagnosis.

Inference Engine: Computer programs that combine data from a knowledge base and a database according to specific algorithms in order to generate a result.

Integrated Database: A filing system for data elements that avoids redundancy in a database.

Knowledge Base Editor: A utility program used to define and store the methods to be followed in order to reach a diagnosis and to provide an interface to the database.

Knowledge Representation: A discipline in which representative elements of a knowledge base are extracted from books, articles, or other documents to produce an integrated information source for facilitating the design of a decision support system.

Massachusetts General Hospital Utility Multi-Programming System (MUMPS): A compact, high-level interpretive data management system designed initially for medical applications. It is particularly suited for interactive applications that require a large shared database and the rapid, efficient manipulation of textual data. MUMPS is an American National Standards Institute (ANSI) programming language used by the Department of Defense, the Department of Veterans Affairs, and many other organizations, institutions, and vendors worldwide.

Modular Software Architecture: Computer software design in which separate parts (modules) perform the different functions of the entire integrated software system.

Neural Networks: A computing system that can process information to automatically derive ('learn') solutions by modeling the processes of biological neural nets.

Posterior Probability: The probability of an event occurring given an initial set of conditions. For example, the probability that a particular disease is present given information on the prior probability of the disease and the conditional probability for the signs and symptoms.

Prior Probability: The expected occurrence rate of an event for a specific population. For example, the expected occurrence rate of a disease in Navy personnel.

Shell Systems: Computer software systems that allow the system behavior to be modified by manipulating data files.

Subjective, Objective, Assessment, Plan (SOAP): A standardized medical format that organizes patient information according to the following categories: **S**ubjective, **O**bjective, **A**ssessment, and **P**lan.

Transition Network: A set of nodes connected by arcs where each connection represents a transition state (not final outcome). This network is traversed by proceeding from one node to another across node-to-node transitions that entail a set of decisions, conditions, properties, and/or constraints.

1.3.2 Abbreviations. The following abbreviations are used in this System/Subsystem Specification (SS).

ADP	Automated Data Processing
AI	Artificial Intelligence
ANSI	American National Standards Institute
AQCESS	Automated Quality of Care Evaluation Support System
BUMED	Bureau of Medicine and Surgery, U.S. Navy
CAMD	Computer Assisted Medical Diagnosis
CAMIS	Computer Assisted Medical Interactive-Video System
CHCS	Composite Health Care System
CNO	Chief of Naval Operations
DBM	Database Management
DNBI	Disease Non-Battle Injury
DOD	Department of Defense
DSS	Decision Support System
FD	Functional Description
IDC	Independent Duty Corpsman
MEDEVAC	Medical Evacuation
MENS	Mission Element Needs Statement
MEPSS	Medical Practice Support System
NAVMASSO	Navy Management Systems Support Office, Chesapeake, VA
NHRC	Naval Health Research Center, San Diego, CA
NMRDC	Naval Medical Research and Development Command, Bethesda, MD
NSMRL	Naval Submarine Medical Research Laboratory, Groton, CT
PHS OMIL	Public Health Service Operational Medicine Informatics Laboratory, Monterey, CA
RAPS	Resource Analysis and Planning System
SAMS	SNAP Automated Medical System
SNAP	Shipboard Non-tactical ADP Program
SOAP	Subjective, Objective, Assessment, and Plan
SS	System/Subsystem Specification
SVE	System Version Editor
TOR	Tentative Operational Requirement
TR	Technical Report (Internal Institutional Publication)
VA	Department of Veterans Affairs

The following are names of existing CAMD systems:

CADUCEUS
DXplain
INTERNIST-1
INTERNIST-1/QMR
MYCIN
ONCOCIN

The following are names of Neural Network programs:

BATCHNET
CASENET

EXSYS is the name of a rule-based Artificial Intelligence program.

SECTION 2. SUMMARY OF REQUIREMENTS

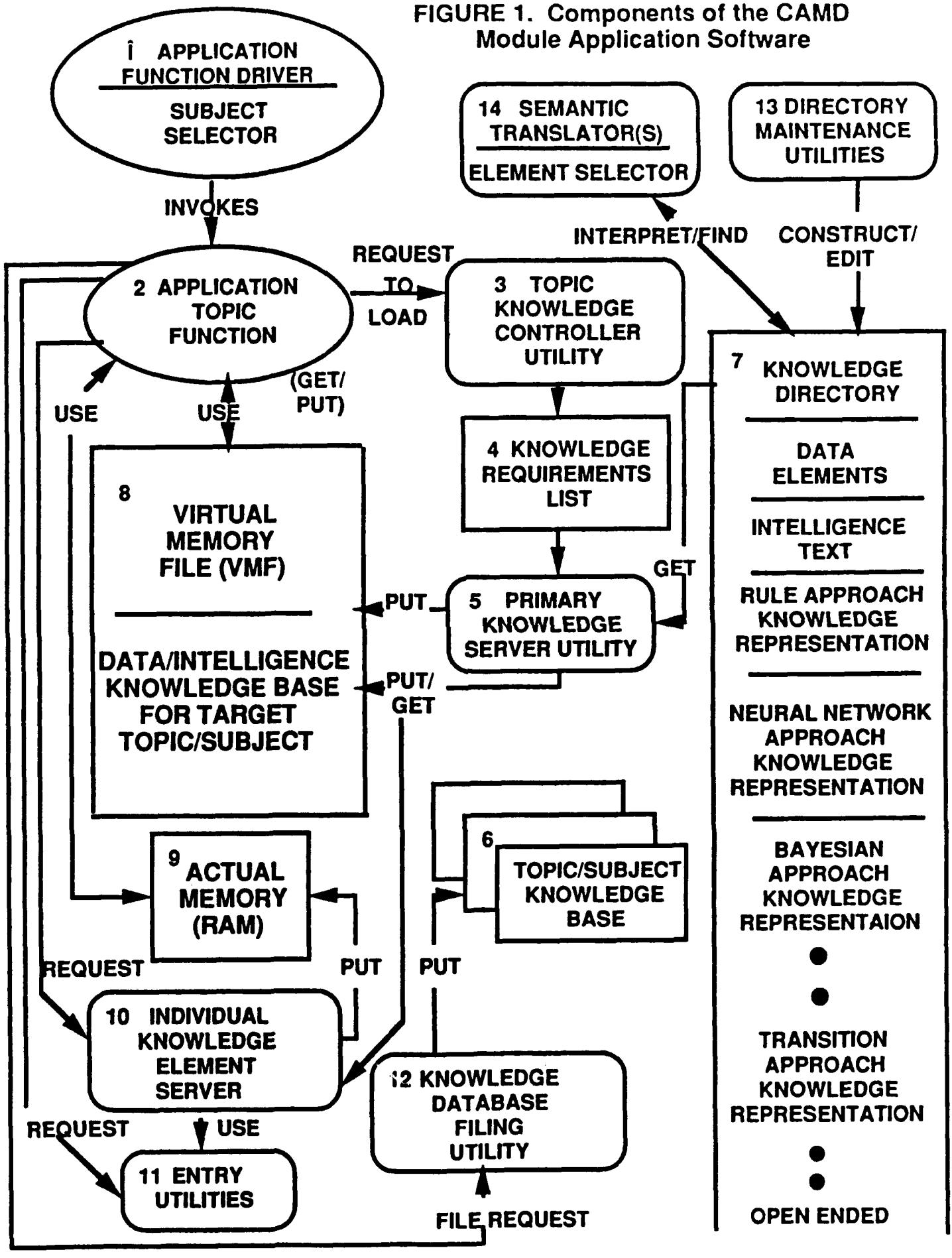
2.1 System/Subsystem Description. The purpose for developing the Computer Assisted Medical Diagnosis (CAMD) module application software in concert with the operational support utility software is to allow bindings to multiple applications written in various programming languages. Without such a facility, it would be extremely difficult if not impossible for these applications to share data and interface with each other. However, with such a facility, multiple applications can be written in any programming language and are regarded as black boxes by the CAMD module operational support utility software which supplies the required interfaces. A general data directory and the operational support utility software will provide continuity of data element structure as well as overall system operation and behavioral consistency for the various functional components of the CAMD module.

Figure 1 depicts the components of the CAMD module application software and system support functions as originally envisioned. The bubbles represent executable routines and routine sets. The boxes or squares constitute data, files, intelligence text, and knowledge representations. In addition to their labels, the bubbles and boxes are numbered for unequivocal identification and reference. The following discussion describes the function of each bubble and the contents of each box.

2.1.1. Application Function Driver/Subject Selector. This component is a skeleton menu function routine set (N routines will drive this bubble). It also will contain a set of routines for selecting the target subject(s) where the subject is equivalent to an individual or entity (e.g., a patient for creating a medical encounter, an Agency to be defined, or an Environment to be worked with). Essentially, the functionality of this bubble allows an end-user to identify the subject on which this transaction is to be done, much like the subject of a sentence.

2.1.2. Application Topic Function. This component allows for the open-ended add-on of any number of black boxes. Each black box represents a particular application written in some programming language. This bubble embodies the logic of the task to be done on the subject selected in component 1 (e.g., in the case of making a differential diagnosis, the inference engine that has as its subject the signs and symptoms information for a particular patient). Examples of topics are an agency unit, mandatory medical requirements associated with an environment, a patient's last encounter, and new lab results. Examples of topic functions are create, add, edit, and display.

FIGURE 1. Components of the CAMD Module Application Software



2.1.3. Topic Knowledge Controller Unit. The application topic puts out a request to load all pertinent knowledge in order to perform that particular topic function. The Topic Knowledge Controller Utility is aware of all of the knowledge requirements necessary to service the specific topic function. For example, it will generate a list of all knowledge elements such as a patient's signs and symptoms in order to perform a rule-based differential diagnosis.

2.1.4. Knowledge Requirements List. This component provides the identification of the knowledge requirements, and outputs the list as a file.

2.1.5. Primary Knowledge Server Utility. The Primary Knowledge Server Utility takes the list from component 4, given the topic function and subject selected, and collect all of the applicable knowledge elements from the appropriate knowledge bases and/or knowledge directory. This utility has awareness of the appropriate knowledge base location for the particular topic/subject knowledge elements.

2.1.6. Topic/Subject Knowledge Base. This component is a collection of knowledge bases relative to the topic/subject (e.g., topic diagnosis, looking in a patient's medical record for his/her signs and symptoms).

2.1.7. Knowledge Directory. The Knowledge Directory is present in the CAMD module software to provide all of the knowledge elements and the control attributes for their interpretation. For example, it provides the knowledge representation for the rule approach (viz, a set of rules syntax) for making a differential diagnosis for any patient. Contained in the Knowledge Directory are data elements, intelligence text and several approaches to knowledge representation (e.g., rule approach, neural net, Bayesian, transition net, and so on). The Knowledge Directory is open ended. New knowledge representations or new data elements can be added to it at any time.

2.1.8. Virtual Memory File (VMF) with Data/Intelligence Knowledge Base for Target Topic/Subject. The Primary Knowledge Server Utility (5) puts all of the knowledge elements and their control attributes received from the Knowledge Directory (7), that is, everything that can be collected for this topic/subject combination, into the Virtual Memory File. Virtual means that there is no limit on the amount of knowledge that can be collected, that is, the Virtual Memory File is unrestricted in terms of capacity. The Application Topic Function bubble (2) then uses this knowledge to retrieve the information it needs to do whatever its about and also to store new information that the topic itself collects in its operation.

2.1.9. Actual Memory (RAM). The Application Topic Function (2) intrinsically uses actual memory (RAM) like any other component program.

2.1.10. Individual Knowledge Element Server. If the Virtual Memory File (8) is lacking a knowledge element it needs, then the Application Topic Function (2) requests the Individual Knowledge Element Server (10) to deliver the missing element to Actual Memory (9). The Individual Knowledge Element Server (10) first looks in Actual Memory (9) to see if it is there. If it is present, its job is done. If it is not present, then it looks in the Virtual Memory File (8) and retrieves the knowledge element if it is there, subsequently putting it in Actual Memory (9). If the missing element is not found in the Virtual Memory File (8), the Individual Knowledge Element Server (10) then sends a request to the Entry Utilities bubble (11) to prompt the end-user for that information.

2.1.11. Entry Utilities. The Entry Utilities component provides all interfacing and prompting functions. The Entry Utilities, given the identification of a knowledge element to be solicited/edited, uses the control attribute information provided by the Knowledge Directory (7) earlier to determine the proper prompting behavior and presents the result(s) back to the Individual Knowledge Element Server (10). This server then completes its work by either delivering the result(s) to Actual Memory (9) and indicating the status (as success or fail, or if a multiple, how many it found) to the Application Topic Function (2), or by putting the result(s) in the Virtual Memory File (8).

2.1.12. Knowledge Database Filing Utility. In any case where new or altered knowledge elements are present, a file request is delivered to the Knowledge Database Filing Utility (12). Only the Application Topic Function (2) can request a filing action. A particular request tells the Filing Utility (12) where the new or altered knowledge element is located, usually in the Virtual Memory File (8). The Knowledge Database Filing Utility (12) has awareness, given the topic/subject combination, of the filing requirements, that is, where in the Topic/Subject Knowledge Base (6) this knowledge element(s) must be stored and any relationships associated with that element(s).

2.1.13. Directory Maintenance Utilities. The Directory Maintenance component contains all of the utility routines necessary to construct or edit any of the basic data element types or intelligence text. This bubble also contains the set of utilities needed to construct and edit each knowledge representation.

2.1.14. Semantic Translator(s)/Element Selector. Semantic translator(s) utilities provide the means to transform certain knowledge elements from internal to externally comprehensible interpretations. This bubble also includes the set of utilities that allows the identification and selection of elements within the Knowledge Directory (7). For example, a graphic image may be stored in digital format that needs to be translated to a dot image required by a printer.

2.2 System/Subsystem Functions. The six major functions of the CAMD module are the following:

a. Diagnostic Support Function. Provide medical diagnosis assistance using information from the medical practitioner and from patient data obtained during previous encounters. This function will provide diagnostic support. Computational techniques may include expert rules; statistical approaches including Bayesian techniques, multiple regression, and discriminant functions; and neural network methods. CAMD should disclose how a diagnosis was made.

b. Therapeutic Plan Function. Support therapeutic plan development. This capability will facilitate proper management of the patient after a diagnosis has been rendered. The function will take into account resources available (e.g., available supplies and equipment) and the circumstances surrounding the event (e.g., ship location).

c. Medical Reference Library Function. Provide a comprehensive source of medical reference data. This capability should contain text, visual material, audio data, and BUMED-approved Navy instructions as appropriate. This function will be accomplished by a smart search engine capable of rapidly locating desired material.

d. Diagnostic Encounter Record Function. Create a computer-stored, patient diagnostic encounter record. This function will access patient demographic data such as sex and date of birth from SAMS, record the date/time stamped differential diagnosis made by the medical practitioner, and pass this diagnostic encounter record to SAMS in the SOAP format.

e. Continuing Medical Education Function. Provide an on-site, computer-based tutoring capability. This function will provide for continuing medical education, including levels of training to match the expertise of the user.

f. Knowledge Base Authoring Function. Create and edit resident knowledge bases. This capability will allow knowledge data to be entered, edited, updated, and retrieved without the author being required to know the file structure. The module will be under total Life Cycle Configuration Management, and diagnostic algorithms will be updated based on verified improved performance.

The CAMD module may employ any of several methods to generate a diagnosis. The various diagnostic methods are the following:

a. **Bayesian Method** - A method for computing the posterior probability of an event from information on the prior probability of the event and associated conditional probabilities using Bayes Theorem: $P(A|B) = [P(A)*P(B|A)]/P(B)$.

b. **Expert Rule Based Method** - An Artificial Intelligence (AI) computer system that consists of an inference engine and one or more knowledge bases. The inference engine examines known data, asks questions to obtain unknown data, and arrives at a probable solution according to a set of IF-THEN-ELSE rules in the knowledge base.

c. **Other Statistical Methods** - Candidate methods are discriminant analysis and both nonlinear and logistic regression analysis.

d. **Artificial Neural Network Method** - Massively parallel computing paradigms that involve many simple processing elements used to derive output values from a set of inputs. Artificial Neural Networks are inspired by neuronal structure and function of the brain.

2.2.1 Accuracy and Validity. The CAMD module must be accurate from both a medical and software standpoint, that is, the diagnostic algorithms must strive to suggest the same diagnoses as those considered correct by medical experts. Verified patient case data will be used to test the validity of the various diagnostic algorithms, and the best performing algorithm will be used for each disease area. As diagnostic algorithms are improved or new algorithms are developed and validated, they will be incorporated into the CAMD module. Electronic libraries for medical reference information and treatment planning will be kept current and as accurate as possible. Library contents will be updated as medical knowledge changes.

2.2.1.1 Diagnostic Algorithms. Diagnostic algorithms will be submitted for validation in the designated format of each knowledge base. Documentation will be provided on the method used to develop and test the algorithms. Test data and an Independent Verification and Validation Plan will also be provided. The algorithms will be reviewed by appropriate medical functional authorities as designated by NMRDC. Approved diagnostic algorithms will be added to the CAMD module by loading the knowledge base provided. Test data will be used to exercise the algorithms before they are deployed as part of the CAMD module.

2.2.1.2 System Data. The CAMD module will validate data as they are entered interactively by the user or as they are received from other automated sources. Validation checks will prevent entry of incorrect or duplicate information. The module will check data for the following:

a. Redundancy. The user will be prevented from creating more than one registration for each patient or posting the same medical data more than once.

b. Integrity. A determination that the CAMD module has a viable database each time it is booted.

c. Input validity. A determination when data are entered that they meet anticipated values or lie within expected ranges.

d. Consistency of external data input. Portions of the CAMD module database come from existing data and external sources. The module will check data from each source for consistency in specific fields such as patient name, sex, date of birth, and social security number.

2.2.2 Timing. The CAMD module will operate in an interactive environment and must provide current and accurate patient data that are easily entered and retrieved. Response time is defined as the time that the computer needs to respond and carry out a user request. Response time is usually measured from the moment that the user presses the Enter Key until the first character of the response is displayed on the screen. Response time requirements are defined below.

a. Interactive Response Time. Interactive processing occurs when a user communicates with computer software in a conversational manner. The computer's operations are monitored directly on a video display so that the user can catch and correct errors before the processing operation is completed. Interactive response time is measured from the moment that the user presses the

Enter Key until the first response character displays on the screen. Interactive response time will be less than 1/4 second for interactive functions. This time does not apply to interface with external systems, generation of reports, extensive file updates, or extensive database searches. The generation of a diagnosis, depending on the complexity of the diagnostic algorithm being computed/interpreted, may take 10-20 seconds. A 20-second upper limit is acceptable.

b. On-Demand Response Time. On-demand processing is the request for hard copy output. On-demand processing includes requests for preprogrammed reports. The CAMD module will return access to the user within 5 seconds of the execution of the request while continuing compilation of the report.

The response time for on-demand output is measured from the time that the user transmits the request until the output is initiated or queued. The CAMD module will be available to the user for the next task within 5 seconds after the request. The output will begin printing or will be queued in less than 5 seconds.

c. On-line HELP Response Time. The response time for on-line help is measured from the time that the user transmits the request until the output is displayed on the user's screen. The response time for on-line help should be less than 2 seconds.

It should be noted that SAMS system response time depends on the internal clock speed of the computer hardware on which SAMS is running. When SAMS is moved to a faster hardware platform, system response times will decrease accordingly.

2.3 Flexibility. The CAMD module will use flexible software and hardware design and architecture to accommodate changing system requirements and to permit interface with other systems. Advanced system design and software engineering technology will be used to provide:

a. A flexible module that can respond to evolving functional requirements.

b. Maximum software reusability and portability.

c. A module that accommodates evolving technology and conceptual changes.

The module uses the concept of modular expansion to accommodate changes in government regulations and advances in technical developments. Modular expansion means that the module

makes use of interim change packages so that total module redesign or replacement is not necessary. The module incorporates a design with application software that is portable to multiple operational platforms. This design also allows capabilities to be extended.

The CAMD module will include a System Version Editor (SVE) that will allow changes, additions, and deletions to the files. This editor will create a version change record that will document all of the changes made by the update. The record will contain the name of the person updating the module, the files edited, and each edit made, line by line, including the original line of code and the changed version line. A printed copy of each of the version updates will be kept by the institution responsible for the updating of the CAMD module. The CAMD module program documentation will be updated to reflect editing whenever new versions of the module are issued or as need warrants.

The SVE will have the ability to incorporate new files including new diagnosis, treatment, or disease information. When new files are added, the names of the people and the institution responsible for developing these new files will be recorded, along with all reports and publications documenting the new files. The new file changes will be incorporated into all program documentation including the user, operator, and programmer guides.

If the CAMD module fails, it most likely will be caused by SAMS itself failing. In the event of a system failure, the alternative courses of action that may be taken to satisfy the information requirements are restoring the latest backup of the SAMS system; rendering a medical diagnosis by the method used before CAMD; and functioning temporarily with degraded modes of operation.

SECTION 3. ENVIRONMENT

3.1 AIS Equipment Environment. The initial version of the CAMD module will operate in the existing SAMS hardware environment. The current SAMS hardware consists of the following:

a. Zenith microcomputer (Z-248) with 512 kilobytes of RAM (640 kilobytes of RAM are recommended).

b. Monochrome or color (CGA through VGA) monitor.

c. Two 20 megabyte hard disk drives (with limited free space).

d. MS-DOS 3.3+ operating system.

Future versions of the CAMD module must support compact disk read-only memory (CDROM) medical libraries and will require the following minimal hardware configuration:

a. 80386 microcomputer (33 MHz) with 640 kilobytes of RAM (upgradable to 4 megabytes of RAM).

b. 5-1/4" and 3-1/2" disk drives (1.2 and 1.44 megabytes).

c. 80 megabyte hard disk drive.

d. 80 megabyte tape backup.

e. VGA or 8514 Color Monitor with 1024 x 768 resolution and 256 colors.

f. Video 5" CDROM.

g. Mouse.

h. Laser printer (at least 4 pages per minute).

i. MS-DOS 4.0 or above operating system with Windows.

3.2 Communications Environment. Communications requirements for the CAMD module itself are minimal. The module will be able to operate on a local area network or on the network used on larger ships such as aircraft carriers. Additional communications requirements are commercially available communications software packages such as ProComm Plus 2.0. No modem is required for the initial implementation of the CAMD module. As the communications requirements for the SAMS system may increase, the CAMD module will be able to take advantage of any expanded communications capability of the SAMS system environment.

3.2.1 Network Description. The CAMD module will be integrated into the SAMS system, and as a consequence, the module will operate within the communications capabilities of the SAMS system environment. Currently, these capabilities are the following:

- Local Area Network
- ProComm Plus 2.0

3.2.2 Physical Interface. The communications hardware required to support the CAMD module will be what exists in the SAMS system hardware configuration. No new hardware acquisition will be dictated by any requirements of the CAMD module. However, as the SAMS hardware platform may be upgraded, the CAMD module will be able to take advantage of the enhanced capability of newer hardware technology.

3.2.3 Protocol Interface. The CAMD module will be developed to operate within the constraints of any SAMS protocol interface.

3.2.4 Applications User Interface. The application user will interface with the CAMD module within the communications environment provided by SAMS at each SAMS operational Navy facility.

3.2.5 Diagnostics. CAMD users will be able to identify and classify any communications problems within the diagnostic procedures provided by SAMS itself.

3.3 Support Software Environment. All CAMD module top level programs should be callable from the SAMS FoxPro 2.0 top level menu and should have the look and feel of SAMS Version 7.0. The operating system will be MS-DOS 3.3+. The CAMD module must use appropriately formatted, existing SAMS patient data. Existing SAMS patient data include demographic information (name, rank, gender, and age), past to current medical history, vital signs, examination findings, and laboratory results. Free-form textual data will not be used for diagnostic decision making because the CAMD module will not have an extensive natural language processor to interpret text.

The CAMD module will undergo user testing at the Naval Health Research Center (NHRC), San Diego, California prior to integration into SAMS. The actual testing will consist of performing 20 tasks designed to fully exercise the CAMD module and verify the ability of this module to be integrated into the SAMS system. The following areas of component and integration testing will be conducted:

- Verifying data links within the CAMD module
- Verifying data links between the CAMD module and the SAMS system

- Entering test sets of actual patient case data into each disease area in the CAMD module
- Generating probable diagnoses for each disease area that accurately reflect the correct diagnosis by expert medical opinion
- Editing and updating patient diagnostic encounters
- Generating the SF600 form in a SOAP format and passing it to the SAMS system

NHRC will be responsible for providing the equipment, facilities, and in-house operators to conduct testing of the CAMD module. After problems and deficiencies revealed by the testing have been resolved or corrected, the CAMD module will be judged acceptable and ready for field testing.

3.4 Software Interfaces. The CAMD module will be integrated into the SAMS system rather than functioning as a stand-alone module. The module in its initial implementation will not interface to any other applications system or subsystem. However, as interfaces between SAMS and other medical information systems may be developed, the CAMD module will also be provided with the capability to interface and exchange patient data between these systems. Existing medical information systems that are candidates for accomplishing such an interface are ACQESS, CHCS, and RAPS. Other medical information systems may become additional candidates in the future.

3.5 Security. The CAMD module as well as the SAMS system itself do not contain any classified information that would require security protection. However, they do contain patient medical encounter data, the privacy of which must be protected. Consequently, the Federal Privacy Act of 1974 applies. ADP security procedures for the medical database files and user identification will be in accordance with SECNAVINST 5239.2. Privacy Act security will conform to the Federal Information Processing Standards Publication 41, Computer Security Guidelines for Implementing the Privacy Act of 1974, 30 May 1975.

3.5.1 Control Points. The control points in the CAMD module where privacy of medical database files must be protected are the following:

a. Transfer of patient demographic and medical history data from the SAMS system to the CAMD module.

b. Access to the patient demographic, medical, diagnostic, and treatment data files within the CAMD module.

c. Access to the diagnostic and therapeutic knowledge bases within the CAMD module.

d. Transfer of patient diagnostic encounter data from the CAMD module to the SAMS system.

3.5.2 Vulnerabilities. Vulnerabilities at each of the control points are the following:

a. Unauthorized access to patient demographic and medical history data transferred to the CAMD module from SAMS.

b. Unauthorized access to the patient demographic, medical, diagnostic, and treatment data files within the CAMD module.

c. Tampering with the diagnostic knowledge bases within the CAMD module that determine how diagnoses are made.

d. Tampering with the therapeutic knowledge base within the CAMD module that provides treatment plans appropriate to diagnoses.

e. Unauthorized access to the patient diagnostic encounter record as it is transferred from the CAMD module to the SAMS system.

3.5.3 Safeguards. The CAMD module will incorporate the following safeguards to protect the privacy of the medical database files:

a. The Security software of the CAMD User Interface Subsystem will provide security for both medical database files and user identification. Users of the CAMD module will be required to enter passwords to gain access to the module. The database files, when they are initially created, will be assigned an attribute making it possible to lock out users.

b. Users of the CAMD module will be notified by on-screen reminders that the files created by the module contain the patient's name, social security number, and any other identifying data elements as well as medical information that is subject to the Privacy Act of 1974.

c. The knowledge bases in the CAMD module will be compiled, making it impossible for users to tamper with them. The only way that they can be changed is by completely replacing an existing knowledge base with an updated, compiled version.

3.5.4 System Monitoring and Auditing. The patient medical database will be a compilation of diagnostic encounter records generated within the CAMD module and then appended to the historical medical encounter record in the SAMS system. As a consequence, ultimate protection of the privacy of the historical

patient medical record, which resides in data files in SAMS and is read-only to the CAMD module, will be the responsibility of the SAMS system itself. All system monitoring will be conducted under the auspices of SAMS operations.

The CAMD module will keep an audit trail of the questions asked of all patients (prompts), all answers to the questions (responses), and the suggested diagnoses. After each deployment, this audit trail will be removed from the CAMD module for analysis by the CAMD module developers to facilitate quality assurance, to improve the accuracy of medical diagnostic support, and to gather additional case data for enhancing the existing diagnostic algorithms. During the time that the audit trail resides in the CAMD module, the privacy of this data file will be protected by locking out users of the module.

SECTION 4. OVERALL DESIGN DETAILS

4.1 General Operating Procedures. The general operating procedures of the CAMD module are governed by the Disk Operating System (MS-DOS 3.3+) command structure and command set.

4.2 System Logical Flow. The proposed data flow among the Independent Duty Corpsman (IDC), the CAMD module, and SAMS is depicted in Figure 2. The CAMD module will provide enhancements to SAMS to help medical practitioners in isolated environments arrive at a differential diagnosis and a therapeutic plan to manage an injury or illness. A database management approach will be used to allow for transfer of patient data from SAMS to the CAMD module and return of a Diagnostic Encounter Record to SAMS.

The components of the CAMD module application software and system support functions depicted in Figure 1 could not be implemented in the FoxPro programming environment as originally envisioned. While it would be possible to implement this original design in either the MUMPS or C programming languages, execution of some of the design features using FoxPro Version 2.0 proved to be either impossible or grossly inefficient because of the characteristics and limitations of FoxPro itself. Consequently, the original design had to be altered to reflect an implementation approach that would be feasible.

Table 1 enumerates the initial implementation components of the CAMD module diagnostic application software and support subsystems. The CAMD module has three basic components: Diagnostic Application Software, User Interface Subsystem, and a Data Directory Subsystem. The Data Directory Subsystem provides core utilities used by the User Interface Subsystem and by application programmers and diagnostic algorithm developers to implement, maintain, and augment the data files within the FoxPro environment.

4.2.1 CAMD Diagnostic Application Software. FoxPro software used for the diagnostic application component of the CAMD module will have five major subcomponents: Patient Encounter, Diagnosis and Treatment Recommendation, Report Generator, CAMD Utilities, and File Maintenance. The functions to be performed by each of these constituents are as follows:

Patient Encounter

- Register Patient Enter/Edit
- Routine Encounter Enter/Edit
- SF600 Enter/Edit
- Evacuation Enter/Edit

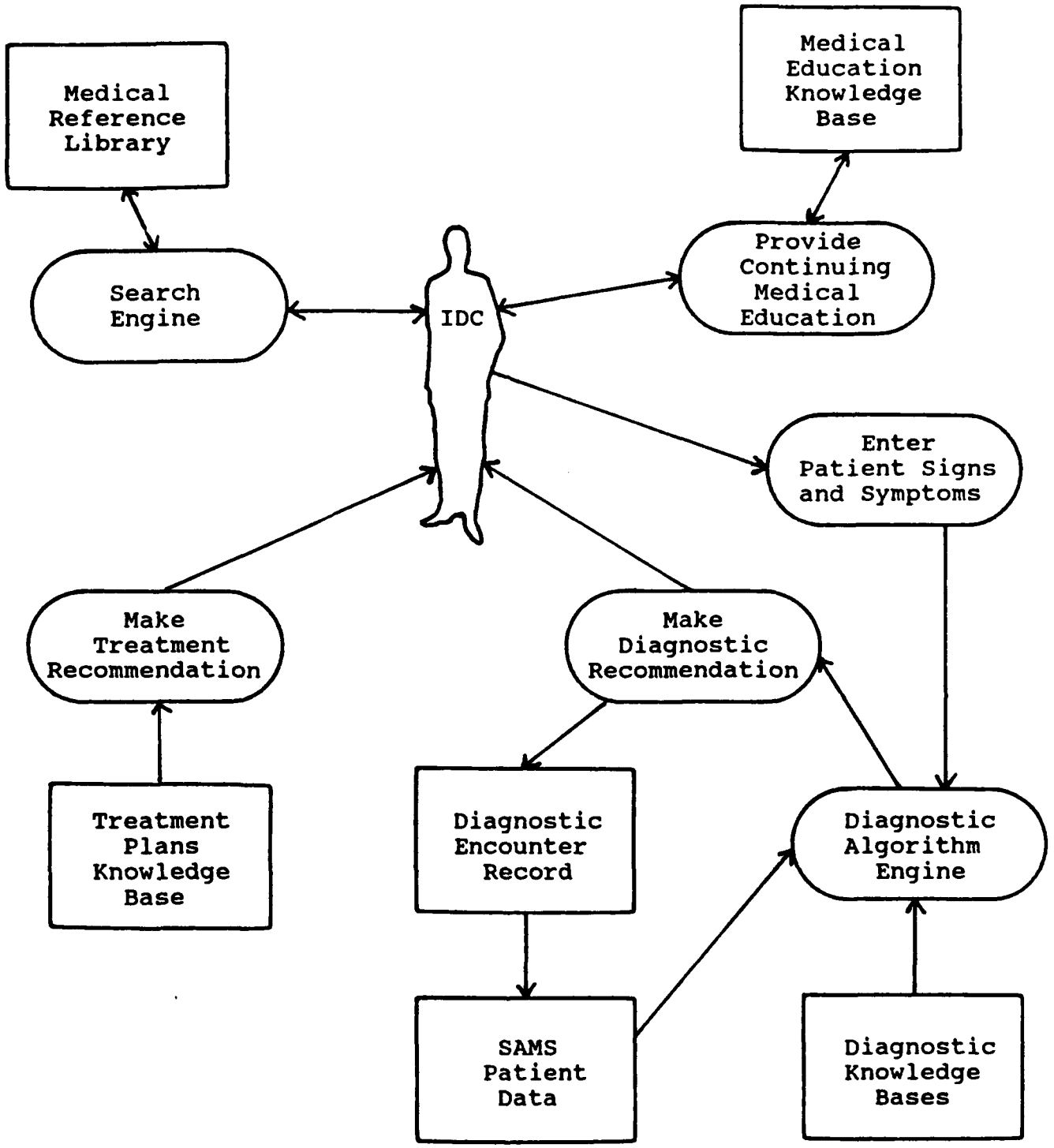


Figure 2. Proposed Data Flow Among the Independent Duty Corpsman, the CAMD Module, and SAMS.

TABLE 1

INITIAL IMPLEMENTATION COMPONENTS OF THE
CAMD MODULE DIAGNOSTIC APPLICATION
SOFTWARE AND SUPPORT SUBSYSTEMS

CAMD DIAGNOSTIC APPLICATION SOFTWARE	**USER INTERFACE SUBSYSTEM**	**DATA DIRECTORY SUBSYSTEM**
Patient Encounter	Security	Create/Edit Directory Entries Utility
Diagnosis and Treatment Recommendation	Window and Screen Handling Utilities	Display Directory Entries Utility
Report Generator	Menu Management and Graphics Handling	Table Entry/Edit Utilities
CAMD Utilities	Procedure Driver	Retrieve Table Entries Utilities
File Maintenance	Help Interface	Data Element Entry and Selection Utilities
	Database Editors	
	Knowledge Base Editor	
	Medical Reference Library	
	Training Scenarios	
	SF600 Form Generator	
	SAMS Interface	

Diagnosis and Treatment Recommendation

- Enter Selected Complaint(s)
- History and Physical Enter/Edit
  (Signs and Symptoms)
- Enter Provider Diagnosis
- Provide Diagnostic Assistance
- Provide Treatment Recommendation

Report Generator

- Ad Hoc Displays (Browse)
- Print SF600

CAMD Utilities

- User ID
- Log-on Security
- CAMD Facility File and Setup
- Archive Patient Records

File Maintenance

- Add/Edit ICD-9-CM File
- Add/Edit Diagnostic Method

In the initial implementation of the CAMD diagnostic application software, Bayesian algorithms will be used to provide a diagnostic recommendation for two sets of signs and symptoms---abdominal pain and chest pain. In the future, three additional areas of diagnostic assistance will be included---ocular, dental, and trauma. Rule-based algorithms for generating a differential diagnosis will be added to the next iteration of the CAMD diagnostic application software.

4.2.2 User Interface Subsystem. Eleven major elements will comprise the User Interface Subsystem: Security, Window and Screen Handling Utilities, Menu Management and Graphics Handling, Procedure Driver, Help Interface, Database Editors, Knowledge Base Editor, Medical Reference Library, Training Scenarios, SF600 Form Generator, and a SAMS Interface. Security will be provided for database files and user's IDs. Window and screen handling utilities will allow the application programmer and diagnostic algorithm developer to use pop up windows or scrolling and to choose the foreground and background color of the screen. The application programmer and diagnostic algorithm developer will also have the capability to manage menus and employ graphics. The procedure driver will further provide the capability to create menus and determine the method of prompting for inputs from the end-user. The help interface will allow the application programmer and diagnostic algorithm developer to author help text for each prompt in the CAMD diagnostic application.

Database editors will facilitate the maintenance and updating of diagnostic application files and tables. The knowledge base editor will allow knowledge data to be entered, edited, and retrieved without the author being required to know the file structure.

A medical reference library can be constructed with this subsystem and accessed using a search engine. Training scenarios can be created and can be driven by the diagnostic engines. The SF600 form will be generated by this subsystem, to be filled in as a SOAP memo by the CAMD diagnostic application software for each patient diagnostic encounter. Finally, this subsystem will provide the interface to SAMS, passing the SF600 SOAP memo to SAMS to use where appropriate. The SAMS database will be read-only to the CAMD module.

4.2.3 Data Directory Subsystem. The Data Directory Subsystem has five kinds of utilities: Create/Edit Directory Entries Utility, Display Directory Data Entries Utility, Table Entry/Edit Utilities, Retrieve Table Entries Utilities, and Data Element Entry and Selection Utilities. Data directory utilities software provides a means for creating all individual data elements and tables of information necessary to support an operational application-specific module such as CAMD. This software will provide identification, definition, and editing utilities for creation of a variety of data elements and associated parameter information necessary to control normal use within an application program or subsystem utility. The software also provides utilities to display directory entries according to several selection methods; to input, edit, and display or print entries in several types of tables; and to execute either data element entry or selection (retrieval) actions.

4.3 System Data. The CAMD module will provide an expert system shell, essentially an empty framework into which developers of knowledge bases and diagnostic algorithms can add medical information, inference engines, rules, examples, and training scenarios. The general categories of information that can be loaded for each knowledge base are described below along with their associated outputs.

4.3.1 Inputs.

a. Bayesian Knowledge Base. The CAMD module knowledge base editor will accept input from developers for the following categories of information:

(1) List of medical diagnoses.
(2) List of signs and symptoms associated with each diagnosis.
(3) List of prompts for the patient's presenting signs and symptoms (questions).

(4) Alternative responses for each prompt (answers).
(5) Prior probabilities for each diagnosis.
(6) Conditional probabilities (weights) for each response alternative for each diagnosis.

b. Expert Rules Knowledge Base. The CAMD module knowledge base editor will accept input from developers for the following categories of information:

(1) List of medical diagnoses.
(2) List of signs and symptoms associated with each diagnosis.
(3) List of prompts for the patient's presenting signs and symptoms (questions).
(4) Alternative responses for each prompt (answers).
(5) The set of rules (e.g., branching tree structure).

c. Classification Functions Knowledge Base. The CAMD module knowledge base editor will accept input from developers of diagnostic algorithms using regression analysis and discriminant analysis methods for the following categories of information:

(1) List of medical diagnoses.
(2) List of signs and symptoms associated with each diagnosis.
(3) List of prompts for the patient's presenting signs and symptoms (questions).
(4) Alternative responses for each prompt (answers).
(5) Prior probabilities (classification function constants).
(6) Classification weights for each response alternative for each diagnosis.

d. Artificial Neural Network Knowledge Base. The CAMD module knowledge base editor will accept input from developers for the following categories of information:

(1) List of medical diagnoses.
(2) List of signs and symptoms associated with each diagnosis.
(3) List of prompts for the patient's presenting signs and symptoms (questions).
(4) Alternative responses for each prompt (answers).
(5) Number of layers in the neural network.
(6) Number of nodes in the neural network associated with each layer.
(7) Pattern of interconnections among the nodes.
(8) Assigned weight for each connection.
(9) Activation function for each node (usually the same for all nodes in a layer).

e. Therapeutic Planning Knowledge Base. The CAMD module knowledge base editor will accept input from developers for the following categories of information:

(1) Treatment descriptions and definitions associated with particular diagnoses.
(2) Drug-to-drug interactions and any counter indications.

f. Medical Reference Library Knowledge Base. The CAMD module knowledge base editor will accept input from developers for the following categories of information:

(1) Medical textbooks.
(2) General medical/surgical journals.
(3) Military medical references such as Manual of Medical Department, General Medical Officer Manual, and Radiation Health Specific.
(4) Navy operational documents such as those pertaining to hazardous materials.

g. Medical Education Knowledge Base. The CAMD module knowledge base editor will accept input from developers for the following categories of information:

(1) Training scenarios for learning how to use the diagnostic support available in the CAMD module.
(2) Training scenarios for learning how to use the treatment planning support available in the CAMD module.
(3) Training scenarios for browsing the electronic medical reference library.

4.3.2 Outputs.

a. Bayesian Method. The following outputs will be produced by application of the Bayesian Method to patient signs and symptoms:

(1) Posterior probability of candidate medical diagnoses in rank order.
(2) In the long term, an explanation of how the candidate medical diagnoses were made.

b. Expert Rule Based Method. The following outputs will be produced by application of the Expert Rule Based Method to patient signs and symptoms:

(1) A medical diagnosis that is the end result of the branching tree rule based structure.
(2) In the long term, an explanation of how the medical diagnosis was rendered.

c. Other Statistical Methods. The following outputs will be produced by application of regression analysis and discriminant analysis classification functions to patient signs and symptoms:

(1) Probability of candidate medical diagnoses in rank order.
(2) In the long term, an explanation of how the candidate medical diagnoses were made.

d. Artificial Neural Network Method. The following outputs will be produced by application of the Artificial Neural Network Method to patient signs and symptoms:

(1) The medical diagnosis generated at the final layer of the neural network.
(2) In the long term, an explanation of how the medical diagnosis was rendered.

e. Therapeutic Planning outputs are the following:

(1) Treatment plans for specific diagnoses.
(2) Medical literature citations from which the therapeutic plan was extracted.
(3) Advice about drug-to-drug interactions and any counter indications.

f. Medical Reference Library outputs are the following:

(1) The results of on-line searches for desired medical information by word, author, subject, or title.

g. Continuing Medical Education outputs are the following:

(1) The results of browsing the electronic medical reference library for desired medical information by word, author, subject, or title.
(2) Delivery of training scenarios for learning how to use the available diagnostic support.
(3) Delivery of training scenarios for learning how to use the available treatment planning support.

4.3.3 Database/Data Bank. The CAMD module shell will provide for the incorporation and manipulation of two databases: (1) a patient database that will use SAMS patient data where available, and (2) knowledge bases that will contain medical reference information and information about how to manipulate the patient data to arrive at suggested diagnoses with their corresponding treatment plans. The CAMD module software will have a set of Data Directory Utilities to provide a means for developers to

create, edit, display, or retrieve the individual data elements and tables of information needed to support the functions of the CAMD module.

In the CAMD module design, a data element can be a numeric value, a character string, a date, a time, a logical value, a free text document, or one of a variety of table configurations. Each data element has a unique Element Identification Code, a Primary Name, and a Data Element Type code such as "C" to indicate a character string. Eight kinds of table configurations will be available such as a "list" or "menu" type of selection table where all selection options (table members) are always displayed as a selection menu. In addition, each directory entry must have a set of control parameter values associated with it that are used to govern data entry or selection processes and other operations involving the data element.

SECTION 5. SUBSYSTEMS DESIGN DETAILS

5.1 CAMD Diagnostic Application Software. When the CAMD module is used for diagnostic support, the module will keep a record of the inputs (key strokes) and outputs of that interaction. This record will include all questions asked of the patient (prompts), all answers to the questions (responses), and the suggested diagnoses provided by the CAMD module. Once a diagnosis has been made by the medical practitioner, the CAMD module will not allow the medical practitioner to remove the signs and symptoms data on which the diagnosis was based.

The audit trail provided by this record will be removed from the CAMD module after each deployment for later analysis. The uses for this audit trail are the following:

a. Provide medical encounter information for quality assurance purposes.

b. Investigate failures of the diagnostic algorithms with the goal of remedying the cause of the failure and improving future performance.

c. Gather additional case data to enhance existing diagnostic algorithms.

The CAMD module will potentially use at least four major diagnostic methods to manipulate the database including the Bayesian Method, the Expert Rule Based Method, Other Statistical Methods, and the Artificial Neural Network Method.

5.1.1 Bayesian Method. A method for computing the posterior probability of an event from information on the prior probability of the event and associated conditional probabilities using Bayes Theorem:

$$P(A|B) = [P(A)*P(B|A)]/P(B).$$

In the case of medical diagnosis determination, the prior probability of the ith disease

$$P(d_i)$$

and the conditional probability of each possible combination of signs and symptoms

$$P(S|d_i)$$

are used to compute the posterior probability of the ith disease

$$P(d_i|S) = \frac{P(d_i)P(S|d_i)}{\sum_{j=1}^{n} P(d_j)P(S|d_j)}$$

where n is the number of different diseases in the problem area.

Finally, because it is assumed that the various features of S (e.g., the array of symptoms, S_k) are independent, $P(S|d_i)$ is computed as follows:

$$P(S|d_i) = \prod_{k=1}^{m} P(S_k|d_i)$$

where m is the number of signs and symptoms.

In the initial implementation of the CAMD module, Bayesian algorithms will be used to provide a diagnostic recommendation for two disease areas: abdominal pain and chest pain. The questions asked of the patient (prompts) and the possible answers to the questions (responses) used by the Bayesian diagnostic algorithm for chest pain are defined in Appendix A, Example of Questions and Possible Answers Used by the Bayesian Method To Suggest Diagnoses for Chest Pain.

5.1.2 Expert Rule Based Method. The CAMD software implemented using this method should be regarded as an expert system. An expert system consists of the following: A rulebase, working memory, a database, and an inference engine.

The **rulebase** is a set of rules for a particular problem area (e.g., abdominal or ocular). Each rule is an independent module consisting of a premise and an action. A rule can fire when its premise is true. When a rule fires, it carries out its action part. Once a rule has fired, it is normally not allowed to fire again, unless the conditions on its input (premise) change.

A rulebase should best be viewed as a causal network, where each rule is a node in the network. Inputs to the rules are the contents of working memory (explained below), and rules also place their outputs back into working memory. The rules should be seen as black box processing elements in the network rather than as procedural code, which is what their textual representation tends to suggest.

The rules are written in some English-like rule language. The description of a rule includes a rule name (or just a rule number), a premise, and the action part:

```
RULE      rule name
IF        premise
THEN      action
```

There are many variants to this basic notation, but the above structure illustrates the important principles involved.

The **working memory** holds the set of symptoms (conditions) and diagnoses for the current session. Initially, working memory is empty. Data elements can be put into working memory by asking the user questions and saving the responses. Data elements can also come from other sources, for example, sensor inputs or case history files. Instead of querying the user, the system obtains the value directly from a sensor or from the case file. When rules fire, their actions can also cause changes to working memory. Typically, rules would alter the confidence factors for diagnoses, but in general, rules can do anything, including adding or removing symptoms. The working memory should be regarded as a collection of active concepts rather than as just passive data storage. When their values change, these concepts are capable of sending messages or triggers to activate rules.

The **database** is a permanent record of the expert system working memory, that is, the set of signs and symptoms and the diagnoses. Usually, the symptoms and diagnoses are copied to the database from working memory at the end of a session. Conversely, to review a session, the symptoms and diagnoses are loaded from the database into working memory. Note that this feature is an extremely important attribute of the expert rule based software implementation. Most expert systems either have no database interfaces or provide an ad hoc, poorly integrated interface capability.

The **inference engine** is a generic rule interpreter that in theory can handle any rulebase. For example, it could be given the name of the ocular rulebase, and it then will run the ocular expert system. Currently, there is a forward chaining inference engine. Backward chaining could also be added, but forward chaining is more suitable for the CAMD diagnostic application. The inference engine searches the rulebase for any rules that are ready to fire. When there are no more rules available, the inference engine terminates the current session.

5.1.2.1 Entity-Relationship Diagram. The Entity-Relationship Diagram is an abstract model of the CAMD diagnostic expert rule based system. The purpose of the model is to show the entities that are to be stored and the relationships between them. The primary goal of the model is to focus on the real world objects, and to avoid implementation details as much as possible. In the interests of clarity, some less important details have been omitted. The model can form the basis for specifying and implementing a database package in any language.

On the Entity-Relationship Diagram shown in Figure 3, entities are depicted as rectangles. Some of the attributes of entities are shown within the rectangle, and the name of the entity is shown in the title bar.

Relationships are represented by lines joining entities. It is customary to name each relationship as a further aid to understanding. Labels have been attached to some of the relationships. Attributes marked with a * are those used to particularize a relationship (commonly referred to as an index in database terminology). A relationship is drawn from the relevant attribute of one entity to the title bar of the other entity. However, relationships can be looked at in either direction. To illustrate, consider the encounter/patient relationship. One could say "encounter X with patient Y," or one could say "patient Y's encounter X." As a final point, relationships can be one-to-one, one-to-many, or many-to-one.

ENTITIES

PATIENT Described by attributes such as name, date of birth, and sex. A patient is uniquely identified by the social security (or other reference) number.

PROVIDER Described by attributes such as name. Uniquely identified by social security number.

ENCOUNTER A patient encounter is a particular event that occurs when a patient (with corpsman/medical practitioner as intermediary) interacts with the diagnostic system. An encounter is identified by the patient, the station, date, and time of occurrence.

DIAGNOSIS The result of an encounter is a diagnosis by the diagnostic system. Describes one or more diseases with associated degrees of confidence.

SYMPTOM During an encounter various symptoms, vital signs, and test results are recorded by the corpsman/medical practitioner. These symptoms fully describe the nature of the patient's complaint.

DISEASE The list of diseases with which the system is capable of dealing. This list will be a subset of all diseases. Each disease can be classified into one or more general areas or categories.

TREATMENT Treatment protocol for each considered disease.

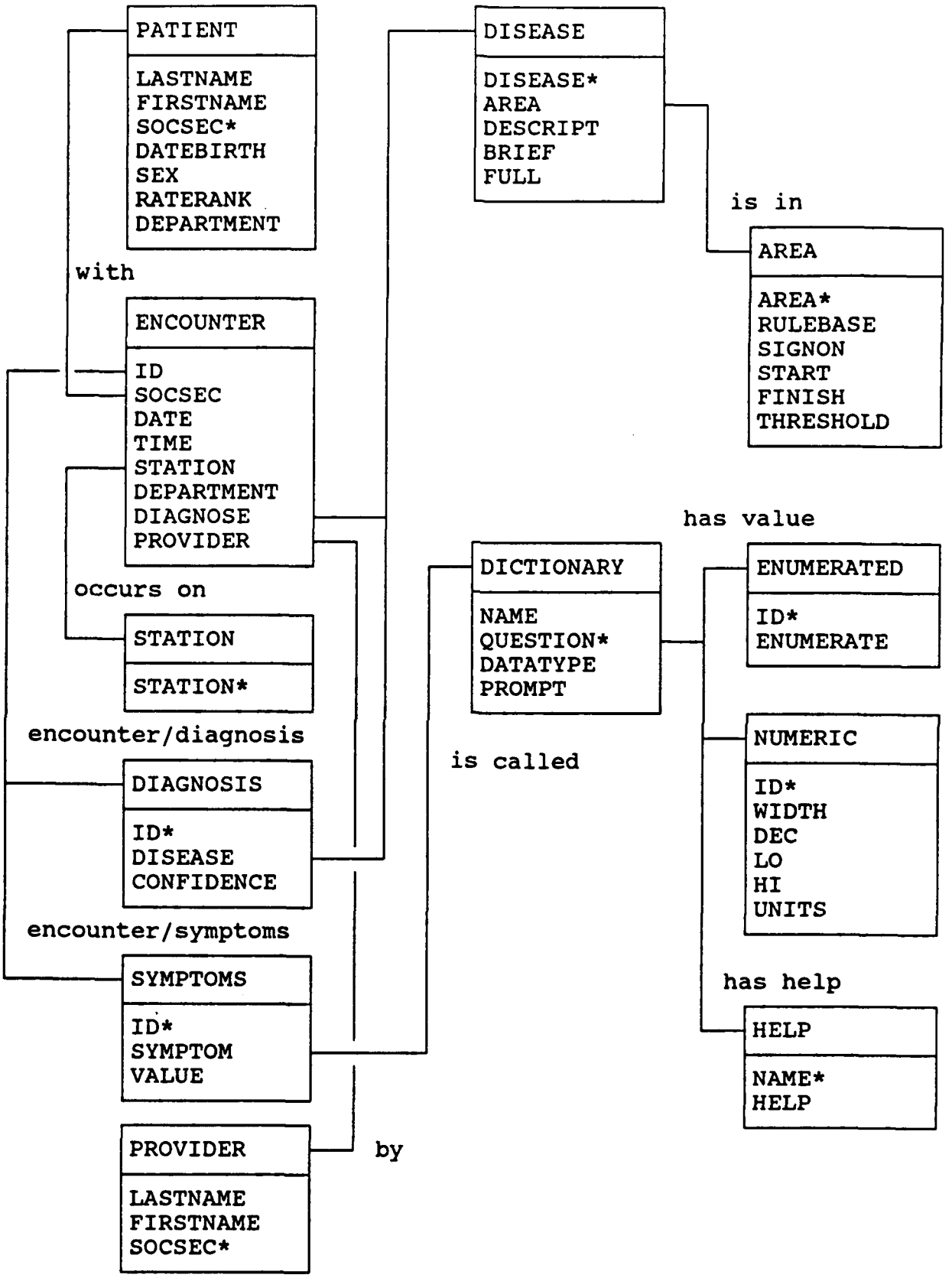


Figure 3. Entity-Relationship Diagram.

RELATIONSHIPS

PATIENT/ENCOUNTER A given patient can have many encounters.

PROVIDER/ENCOUNTER A provider can have many encounters with many patients.

ENCOUNTER/DIAGNOSIS An encounter can have one or more diagnoses associated with it.

ENCOUNTER/SYMPTOM An encounter has several (many) symptoms recorded for it.

DIAGNOSIS/DISEASE A valid diagnosis must be for a disease in the disease catalog.

SYMPTOM/DICTIONARY A valid symptom must be one found in the symptom dictionary.

DISEASE/TREATMENT A disease can be accompanied by one or more treatment protocols.

DISEASE/AREA A disease can be classified as falling primarily within a certain category or diagnostic area (e.g., abdomen, chest, dental, ocular, trauma).

DICTIONARY/VALUE An item in the dictionary is fully described by a set of values that it can have.

5.1.2.2 Expert Rule Based Database Implementation. The detailed database structures currently used are described below. Note that these are very flexible structures that can easily be changed to meet new requirements without a major impact on the software. In particular, the length of character fields is arbitrary. A nominal length of 30 characters has been chosen in most instances, but this could be different in a later version.

Wherever possible, a data-driven philosophy has been adopted to minimize the dependence of program logic on particular instances of data. As the prototype is refined, the necessary database structures will be added to fully implement data independent code.

The application program and the database interface module will be separate .EXEs. Database interface is loaded as TSR, accessed by interrupt vector. The database interface module needs to provide high-level database services to the FoxPro program.

The design objective is to separate user interface concerns from the database and knowledge base tasks. The aim is to reduce maintenance by allowing the user interface to be easily changed without having to change the database component.

Another objective is to allow different modules of the program to be handled independently in such a way that, once the module conventions have been specified, different groups of people can work on separate modules, without excessive communications between the groups. This capability will be facilitated by separately compiled modules under MS-DOS in which modules exchange messages. To make this work effectively, each module needs to be object oriented, that is, each module provides services without needing any knowledge about the internal details of any other module.

FIELDS

The fields of each database are listed in the order in which they appear in the file along with field type, field width, and number of decimal places (where applicable to numeric fields). The following abbreviations for field types are used (consult the FoxPro documentation for more details):

C Character
N Numeric
L Logical
M Memo
D Date

INDEXING

Indexed fields are shown with the accompanying index file names next to the applicable field. For example, the database BAYES.DBF is indexed on the field CONDITION to the index file BAYES.NDX. In most cases, unless otherwise specified, character index fields use capital letters. This means that the words "APPENDICITIS" and "Appendicitis" are not the same! If the first appears as an indexed term in a database, then an attempt to search for the second will fail.

In some cases, more than one database field is involved in the index expression. Where applicable, the index expression is shown alongside the index file name, and the index points to the aggregate of fields.

READ-ONLY DATABASE TABLES

AREA.DBF			
AREA	N	4	
NAME	C	30	
RULEBASE	C	10	
SIGNON	C	30	
START	C	30	
FINISH	C	30	
THRESHOLD	N	6	2
PROBABLE	N	6	2
LIKELY	N	6	2
RULES	N	4	

AREA.NDX

Diagnostic area dictionary, indexed by area name. This file is used to access all supported diagnostic areas. Each area corresponds roughly to a differential diagnosis. The AREA field is the identifier for the diagnostic area. It is used internally to select a set of diagnoses for consideration by the diagnostic algorithm. The NAME field is the (user oriented) name for the diagnostic area or complaint. The THRESHOLD, PROBABLE, and LIKELY fields are display thresholds for the diagnoses. This feature allows three levels of confidence in the computer's diagnoses to be presented to the user. A factor with a value less than the threshold is considered insignificant and is inhibited from appearing in displays and reports.

AREA is the name of a major diagnostic area or subspeciality.

RULEBASE is the name of the rulebase for this area. This rulebase will be used to guide the diagnosis session when this area is selected.

RULES is the number of rules in the rulebase for this area.

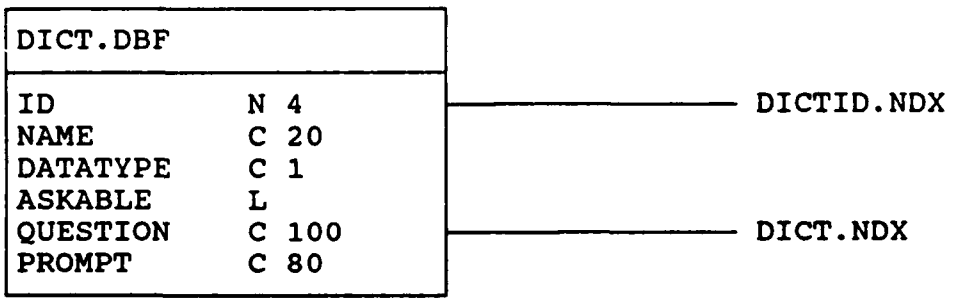
SIGNON is the name of the signon screen displayed when this session first appears. START is the name of the screen displayed before the session begins, and FINISH is the name of the screen displayed before the diagnosis is presented to the user.

COLORS.DBF		
NAME	C	10
COLOR	C	12

Color set. This unindexed file contains all of the configurable screen and window colors available to CAMD. There can be multiple instances of these color sets available so that different ones can be loaded and accessed at different times, depending on requirements.

NAME is the name of the configurable color variable, used by the CAMD expert rule based system at run-time to set the current color. COLOR is the specification of the foreground, background, and highlighted colors, following the FoxPro syntax. For example:

W+/N,N/W specifies high intensity white foreground, on a black background, with highlighted items shown as black letters on white.



This table describes each data element referenced by the diagnostic algorithm, the list of signs and symptoms used by the system, and a specification of the possible values and types that the symptoms can have. This file is used to look up and validate signs and symptoms during encounters or diagnoses.

ID is the primary identifier of the symptom. The ID field is used to select the item from the symptom dictionary.

The NAME field is the (user oriented) short name for the data element. Examples are "TEMPERATURE" and "DIASTOLIC BP."

The ASKABLE field determines whether the user can be asked to enter a value for the item. Nonaskable items, known as meta-qualifiers, are generally computed values that may not have any meaning to the user. The system itself must derive the value from other known quantities.

The QUESTION field is used to ask the user for the value of the data element if it is missing. The question is triggered automatically by the diagnostic algorithm when it tries to evaluate a clause.

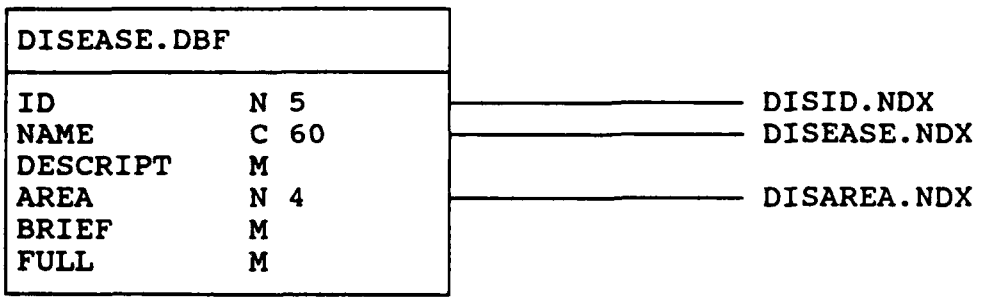
DATATYPE is the data type of the symptom. It can be one of the following types:

E Enumerated type. Symptom can take one of a list of possible string values (see ENUM.DBF).

M Multiple valued enumerated type. Similar to E above, but symptom can have several values at once.

N Numeric type. Symptom can take on a numeric value with a specified range and precision (see VAL.DBF).

L Logical type. Symptom can take on true or false values (or any of the variants, such as YES or NO).



This table describes all diseases that the system can handle. It is used to retrieve disease descriptions, treatment descriptions, and to construct differential diagnoses. The diagnostic engine will select a subset of the diseases in this table in generating a diagnosis.

The ID field is a unique identifier for the disease. It is used by the diagnostic engine to select diseases for the differential diagnosis.

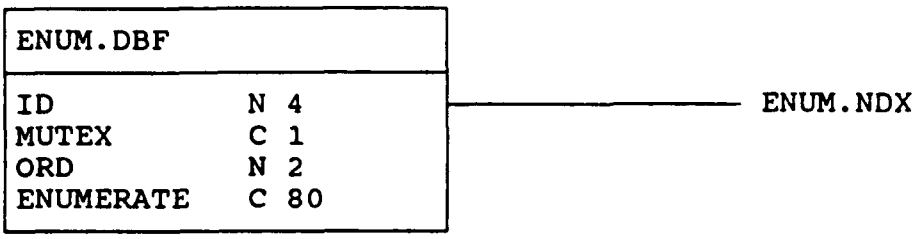
The NAME field is the (user oriented) external name for the disease. It is used to display or print results only.

The DESCRIPT field is the (user oriented) disease description. It is used to display, for the user, information about the disease and its treatment.

BRIEF is a brief treatment description, and FULL is a more comprehensive treatment description.

All descriptions are stored as memo fields that can be accessed or edited using a text editor. See also HELP.DBF.

Accessing the file using the AREA as key allows all diseases associated with that area or differential diagnosis to be retrieved.



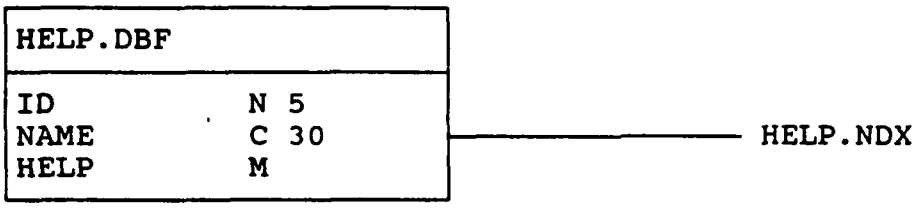
This table lists the enumerated values for data elements of type Enumerated or Multiple value.

The ID field corresponds to the DICT.DBF dictionary ID referenced by this item. Each dictionary item can have several Enum choices associated with it. The ENUM.DBF dictionary is indexed on the ID field. This indexing allows all possible values for a given symptom to be retrieved, using the symptom name as key in the DICT file.

The ENUMERATE field is the actual value of the element.

The ORD field is a subelement identifier or ordinal for the enumerated value.

The MUTEX field is used for multiple selection types in order to determine mutual exclusion between certain user selections. Each symptom can have one or more string values associated with it, that is, the symptom can take on more than one (inclusive) value at one time. All (inclusive) string values for the symptom are shown with MUTEX field +. Some values of the symptom cannot occur simultaneously with any others, that is, if the symptom takes on this value, then all other possibilities must be excluded. All exclusive string values for a symptom are shown with MUTEX field -.



Online help file. The NAME field is the name of an item for which help text is recorded in the HELP memo field. Free form text of arbitrary length appears in the memo field and can be edited, printed, or displayed using any text editor. Items can be created by adding their names to the index and placing the text in the memo field. These items can only be accessed by referencing the index term somewhere in the program. The length and content of the actual text in the help file can be edited or changed at any time. The revised text is automatically invoked

when the system user calls up the help screen.

CAMD.PCX

CAMD signon display screen. This file is a PCX-format file that is displayed in a graphics mode on EGA or VGA screens.

RULE.DBF		
RULE	N	5
AREA	N	4
SALIENCE	N	3
PREMISE	N	5
ACTION	N	5
TEXT	M	

RULE.NDX

Expert system rulebases, indexed by rule name. Each rule is uniquely named so as to belong within a certain diagnostic area (see AREA.DBF). Each rule ID further allows the rule to be retrieved by rule number, within the area. This scheme allows multiple rulebases to be simultaneously accessed online.

PREMISE.DBF

This table is the set of all clauses to be evaluated by the rules. Each rule can select several premises. All premises associated with the rule must evaluate to true in order for the rule to become eligible (placed on the agenda). Therefore, there is an implied AND operator joining the clauses. A premise consists of the following fields:

clause	premise identifier
op	operation code
object	object type descriptor
id	object identifier
val	value

The clause field is used to select the set of clauses associated with the rule.

The op field is the operation code (relational operator) that is to be applied.

The object field describes the type of object pertaining to the clause. Currently, this type can be either an "S" for sign/symptom or "D" for disease. This object descriptor allows the appropriate sign/symptom or disease table to be selected for evaluating the clause.

The id field is the identifier of the particular object referenced by the clause. This id is used to retrieve the item from the dictionary or disease table.

The val field is the value against which the item will be evaluated.

NOTE: A premise is a simple expression with the following syntax:

<objectid> <op> <val>

where

<objectid> is the name of a symptom or disease (as described in DICT.DBF and DISEASE.DBF)

<op> is the operator which can be one of the following:

```
<
>
==
!=
<=
>=
=
```

<val> can be any number or string constant (string constants are defined in ENUM.DBF).

Examples:

```
TEMPERATURE > 102
COLOR == "PALE"
```

This is a very simple syntax, but it is sufficient to handle the type of expressions required in the Bayesian rulebase. Note that because of the implied AND, the syntax will support expressions of the form:

```
TEMPERATURE > 102 AND COLOR == "PALE"
```

ACTION.DBF

This action table is very similar to the premise table. The action clauses are evoked when the rule is selected from the agenda. Each action clause results in some operation being applied to the working memory, whereas premise clauses usually test the values in working memory. Apart from this difference, the structure of the two clause types is almost identical. An action consists of the following fields:

clause	action clause identifier
op	operation code
object	object type descriptor
id	object identifier
val	value

The clause field is used to select the set of action clauses associated with the rule.

The op field is the operation code that is to be applied to the object.

The object field describes the type of object pertaining to the clause. Currently this can be either an "S" for sign/symptom or "D" for disease. This object descriptor allows the appropriate sign/symptom or disease table to be selected for evaluating the clause.

The id field is the identifier of the particular object referenced by the clause. This id is used to retrieve the item from the dictionary or disease table.

The val field is the value against which the item will be evaluated.

NOTE: An action clause, similar to a premise, is a simple expression with the following syntax:

<objectid> <op> <val>

Examples:

COLOR = "PALE"
APPENDICITIS = 58.3

VAL.DBF		
ID	N 4	VAL.NDX
WIDTH	N 2	
DEC	N 1	
LO	N 9 2	
HI	N 9 2	
UNITS	C 10	

This table describes the attributes for numeric types.

The ID field corresponds to the dictionary id referenced by this item. Each dictionary item can have only one val associated with it.

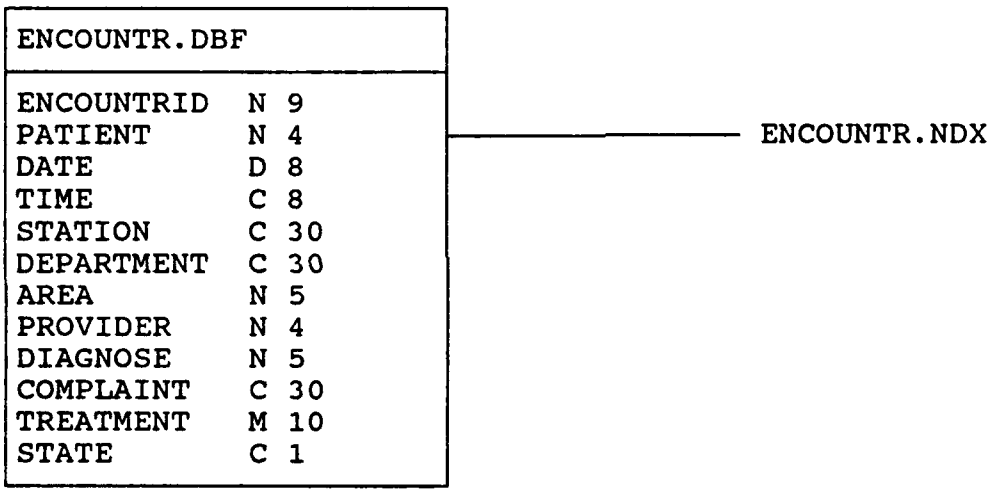
The WIDTH field describes the total width allocated to this data element. This limits both storage as well as display of this item.

The DEC field is the number of decimal places of precision allowed for this item.

The LO and HI fields, respectively, give the lower and upper range of this data element. This range is used to validate the element when entered by the user or imported from an outside source. Either LO or HI left blank implies that there is no range check applied.

The UNITS field describes the (user oriented) engineering units for the physical quantity represented by the item (e.g., degrees centigrade, mm Hg, inches, lbs.).

TRANSACTION FILES



List of primary patient details for a given encounter.

The ENCOUNTRID field is the unique identifier of an encounter.

PATIENT id is the patient identifier, used to look up the patient details in the patient file.

DATE and TIME record the date and time of the encounter.

STATION is the name or code of the ship or station of the patient at the time of the encounter.

DEPARTMENT is the designated department to which the patient is assigned.

AREA is the name of the primary diagnostic area selected by the medical practitioner or by the system when making a diagnosis for this encounter.

PROVIDER is the identifier of the medical practitioner (user) that was logged on to the system when the encounter was recorded.

DIAGNOSE is the diagnosis entered by the medical practitioner for the encounter.

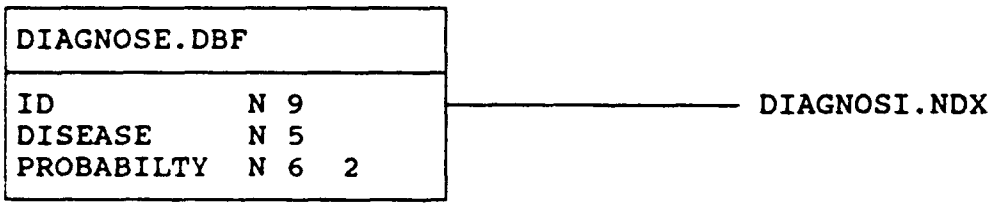
TREATMENT is the treatment entered by the medical practitioner in free text form.

COMPLAINT is a free form (currently) text field that describes the patient's complaint (reason for the visit).

STATE is the state of the encounter. An encounter can be in one of two states:

S Suspended
C Closed

The file is indexed on the PATIENT field. This indexing allows all encounters for a given patient to be retrieved. Symptoms and diagnoses for a given encounter then can be retrieved using the unique encounter key.



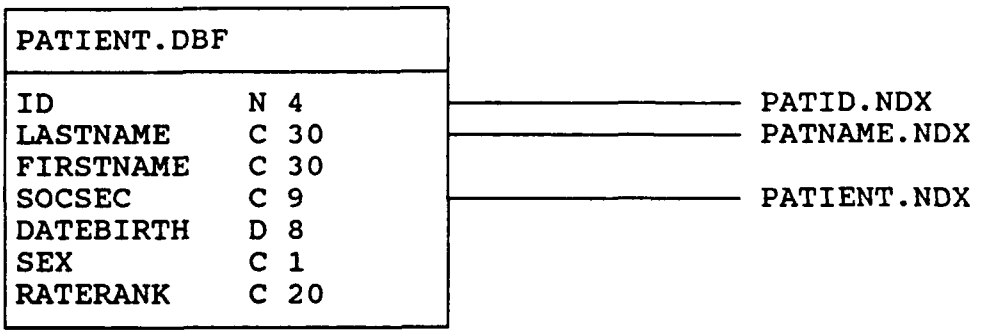
List of computer-generated diagnoses for a given encounter.

The ID field is the unique identifier of an encounter.

DISEASE is the name of the disease recorded for this encounter.

PROBABILITY is the calculated probability of occurrence or confidence factor for that disease.

The file is indexed on the ID field. This indexing allows all diagnoses for a given encounter to be retrieved by a single key. The key is obtained from the ENCOUNTR.DBF file.



List of known patients.

The LASTNAME and FIRSTNAME fields record the names of the patient, LASTNAME being the (possibly nonunique) primary key for accessing a patient by name.

SOCSEC is the unique social security or other reference number by which the patient is known externally.

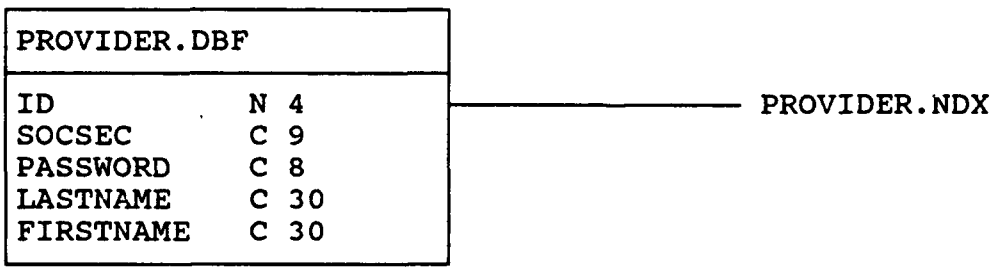
ID is the system level identifier uniquely and unambiguously accessed in the patient database.

DATEBIRTH is the patient's date of birth in date format.

SEX is the sex of the patient recorded as M (for Male) or F (for Female).

RATERANK is the patient's military or other rate/rank designation.

DEPARTMENT is the name of the department to which the patient is assigned.



List of known providers.

The LASTNAME and FIRSTNAME fields record the names of the provider, LASTNAME being the (possibly nonunique) primary key for accessing a provider by name.

SOCSEC is the unique social security or other reference number by which the provider is known externally.

ID is the system level identifier uniquely and unambiguously accessed in the provider database.

SYMPTOMS.DBF		
ID	N 5	SYMPTOMS.NDX
SYMPTOM	N 4	
VALUE	C 10	

List of signs and symptoms entered by the medical practitioner for a given encounter.

The ID field is the unique identifier for an encounter.

SYMPTOM is the name (DICT ID) of the sign or symptom recorded for this encounter.

VALUE is the value taken on for that symptom.

The file is indexed on the ID field. This indexing allows all signs and symptoms for a given encounter to be retrieved by a single key. The key is obtained from the ENCOUNTR.DBF file.

SYSTEM.DBF	
PASSWORD	C 10
PATIENTS	N 4
PROVIDERS	N 4
ENCOUNTERS	N 4
STATION	C 30
DEPARTMENT	C 30
DEBUG	L 1
CONF	L 1
SHOWRULE	L 1
NOVICE	L 1
EDITHELP	L 1

System file. Used to track transactions and control access.

PASSWORD is for the system administrator, initially the only authorized user.

PATIENTS, PROVIDERS, and ENCOUNTERS list, respectively, the number of entities of each type that have been entered into the system. For example, when a patient is added to the system, the

PATIENTS counter is incremented.

STATION. This field will be unique for each installation. The obvious parameters are the name of the ship or station where the system is deployed.

DEBUG controls debug mode. This flag is only used by system programmers for test purposes. During normal system operation, it will be FALSE.

CONF controls display of confidence factors. If it is TRUE, confidence factors are shown for disease diagnoses.

SHOWRULE controls display of rules as they are fired. During normal operation, this will be FALSE, and no rules are displayed.

NOVICE controls novice/advanced system operation. In novice mode, extra help prompts, hints, and guidance are displayed to the user. If this flag is FALSE, these prompts are not shown.

EDITHELP controls editing of on-line help files. If this flag is TRUE, the user is allowed to edit and save help screens while running the system. This facility is intended to be used by maintenance and system programmers in order to tailor the help system to the particular needs of the user.

5.1.2.3 Expert Rule Based System Documentation. Additional detailed documentation is provided for the CAMD Expert System main menu and screens in Appendix B. The knowledge base and procedures for maintaining the rulebase and associated data structures are described in detail in Appendix C. Appendix D describes the CAMD Expert System software test plan in detail.

5.1.3 Other Statistical Methods. Nonlinear and logistic regression, and discriminant classification functions will be supported. For each of these statistical methods, the CAMD module knowledge base editor will incorporate the following information, provided by the diagnostic algorithm developer, to calculate a predicted diagnosis:

(1) List of medical diagnoses.

(2) List of signs and symptoms associated with each diagnosis.

(3) List of prompts for the patient's presenting signs and symptoms (questions).

(4) Alternative responses for each prompt (answers).

(5) Prior probabilities (classification function constants).

(6) Classification weights for each response alternative for each diagnosis.

(7) Probability of candidate medical diagnoses in rank order.

5.1.4 Artificial Neural Network Method. Massively parallel computing paradigms that involve many simple processing elements used to derive output values from a set of inputs. The CAMD module knowledge base editor will incorporate the following information, provided by the diagnostic algorithm developer, to arrive at a predicted diagnosis:

(1) List of medical diagnoses.

(2) List of signs and symptoms associated with each diagnosis.

(3) List of prompts for the patient's presenting signs and symptoms (questions).

(4) Alternative responses for each prompt (answers).

(5) Number of layers in the neural network.

(6) Number of nodes in the neural network associated with each layer.

(7) Pattern of interconnections among the nodes.

(8) Assigned weight for each connection.

(9) Activation function for each node (usually the same for all nodes in a layer).

(10) The medical diagnosis generated at the final layer of the neural network.

5.2 User Interface Subsystem.

5.2.1 Directory Interface Utility Routines.

Function ZUTIP (mapp, mdefault, @mrdata, @mrkey)

Purpose: To provide a general entry utility for nontable type directory codes and a selection utility for Directory table types TA, TB, TL, TM, and TP. No end-user access to table types TC or TN is allowed. The ZUTIP utility is the primary routine used for the entry of general directory code elements and the selection of entries from all user-accessible table type elements.

Parameters:

mapp: Directory EIC of the table.

mdefault: For nontable type codes, mdefault can be either null or must contain a default value having a FoxPro data type corresponding to the directory code data type for Character, Date, Numeric, or Logical. Directory element types C, D, N, and L directly correspond to the FoxPro data types. All other directory types---M, I, and R---are treated as Character data types. Type M element default values can also be a memo field, which in FoxPro is also treated as a Character string. If mdefault is null, the ZUTIP utility will "set" mdefault to the "initial" value for the appropriate data type as follows:

null for elements treated as Character type data.
" / / " for Date elements.
0 for Numeric elements.
.F. for Logical elements.

For TA, TB, TM, and TP table types: The mdefault variable contains either the search criteria, an existing response key or name value, or a null (empty) value as described below. If mdefault is null and the table does not utilize a "list" type of selection (parameter zdisp is set on), this routine will prompt for the search criteria using the applicable zpmt or zname directory parameter text.

For TA or TM tables: If the zdisp parameter is off, mdefault must be null or must contain either an existing table entry name value or a search criteria. If the zdisp parameter is on, mdefault must be either null or contain an existing table entry name.

For TB or TP tables: If the zdisp parameter is off, mdefault must be null or must contain an existing table entry key or name value, or a search criteria. If the zdisp parameter is on, mdefault must be either null or contain the key value of an existing table entry.

For TL table types that allow only one selection, mdefault is the initial selection setting and must be either zero (no initial selection), a numeric value within the range of option numbers, or null (no initial selection). For TL tables that allow multiple selections, mdefault may be either null or must contain a string of 0 (not selected) and 1 (selected) characters that correspond to each table option number in a left-to-right sequence beginning with selection option 1. A null value will be treated as an all zero (none selected) string.

mrkey: This parameter does not need to be present for any nontable type of directory code. When present, it must be sent by reference and should have an initial null value. For successful selection from TA, TB, TC, TM, TN, and TP table types, it is returned set to the ZID field (key-value or index-value) value of the selected table entry in the TAB.DBF for table types TA, TB, TL, and TP or the TAM.DBF file for table types TC, TM, and TN. For single selection TL tables, it is returned as a numeric value, set to the option number of the selection or set to 0 if no selection was made.

For TL tables that allow multiple selections (multiple selection lists and check boxes), mrkey is returned set to a string of 0 (not selected) and 1 (selected) characters corresponding in left-to-right order to the table options.

mrdata: This parameter must be sent by reference and should have an initial null value. For nontable data types, mrdata is returned set to the response value entered or to the default value if no entry or alteration was made. If an existing value was deleted, mrdata is returned empty for character type data, or set to the "empty" value that corresponds to the FoxPro conventions for this condition for other data types.

For successful selection from TA, TB, TM, TP, and single selection TL tables, mrdata is returned set to the "NAME" field (name or text value) of the selected entry or option.

For TL tables that allow multiple selections, mrdata is returned set to an "^" delimited string wherein each "^" piece corresponds in left-to-right order to the table options and to the sequence of 0 and 1 characters returned in the mrkey variable. When a 1 value (denoting selection) is present in mrkey, the name/text value of the relative option number is inserted in the corresponding "^" piece of the mrdata variable. When a 0 (denoting not selected) is present in mrkey, the "^" in mrdata that corresponds to that option will be null (empty). For example, if options 2, 3, and 5 of a 5-option list were selected, mrkey and mrdata would be returned as follows:

```
 mrkey:  "01101"
mrdata:  "^Op2-name^Op3-name^^Op5-name"
```

NOTE: The ZDP function can be used to extract any "^" piece value from mrdata.

Returned value: For nontable type element entry/edit actions, the function value is returned as follows:

0 = No alteration or entry action was done. The variables mdefault and mrdata are identical.
1 = mrdata contains either a new entry or a response different for the given existing default value.
2 = The existing response was deleted (erased from the response entry field).

NOTE: In no event will a non-null default value given in parameter mdefault be altered by this utility. A null default value given for a Date, Logical, or Numeric element will automatically be converted to a valid type configuration.

For table type element selection actions, qf is returned set to 0 denoting a successful selection action. Any value greater than 0 denotes an aborted selection, and any value returned in either mrkey and/or mrdata is to be ignored. The mrkey and mrdata parameters are set as described above for successful selection actions.

Operation: This routine assumes it has been invoked in a window that is suitably dimensioned for either general use with any table selection action or a selection action specific to the subject table. For TL tables that are "check boxes," "push buttons," or "radio buttons," the window must be large enough to accommodate all options in no more than two columns. The routine will use single rows or columns for options when window space allows. The selection action or, if applicable, the search criteria prompt will begin at row 0 column 3 of the active window. An example of a window definition suitable for use in this routine and for other general directory code entry/selection is as follows:

DEFINE WINDOW xxx FROM 10,5 TO 23,74 DOUBLE FLOAT COLOR SCHEME 5 (Dialog) or Scheme 18 from DONS color set.

The window used must be large enough to hold all prompting text, the response area, and any additional lines used for spacing by the entry routine. If memo data are used as the prompt, the user must consider the final line length of memo data.

NOTE: It is critical that the variable type (Numeric or Character) for the mdefault parameter be correct for the default value if it is not null. All single selection TL table actions require mdefault to be a numeric value and return mrkey as a numeric value. In all other cases, mdefault must be a character type, and mrkey is returned as a character string. The mrdata value is always returned as a character string regardless of the content.

Environment:

TALK must be OFF.
DELETED must be ON.
Current Database selection at call is restored at exit.
Windows used: ZUW_disp and ZUW_msg.
Uses color schemes 18 and 20 from set "DONS" in resource file ZRSRC.DBF.
Calls: Routines ZUTAL for selection from TL type tables and ZUTAB for selection from all other tables.
Databases: May use any or all Z*.DBF files.

Function ZUTABR (mapp, mtype, mdkey, mscrl)

Purpose: To display the contents of an existing directory table.

Parameters:

mapp: Directory EIC of the table.

mtype: Table type (TA, TC, TB, TL, TM, TN, TP).

mdkey: Applicable only to TB and TP table types.

0 = Do not include key values with each entry name value in the display.

1 = Include corresponding key values with each entry name. The key values will be enclosed within brackets. For TB tables, the key will precede the name in the display. For TP tables, the key will be inserted after the name value and before any additional identification data.

mscrl: Not applicable to TC, TM, and TN table types.

0 = Table contents are to be displayed in a scrolling window without use of a memory array.

1 = Table contents are to be displayed from a memory array as a list. This value is always used for TL tables. The user must assure that there is sufficient memory space available to temporarily hold all table entries.

Returned value: Returns logical True.

Operation: For table types TC, TN, and TM, a "Browse" window is used to allow selection of any table entry and inspection of the memo field contents. All other tables are displayed within an automatically sized window. Modification of table content is disabled in this utility. The display order is the definition order for TA, TC, TL, TM, and TP tables, and key value order for TB and TP tables.

Environment:

TALK must be OFF.
DELETED must be ON.
Current Database selection at call is restored at exit.
Windows used: W_brz and WZ_disp.
Uses color schemes 19 and 20 from set "DONS" in resource file ZRSRC.DBF.
Calls: Routines ZULST and ZUTLD.
Databases: May use any or all Z*.DBF files.

Function ZUTLD (mapp, mn, @mbar, @mkey)

Purpose: To load the key and/or name values of a directory table into memory array(s).

Parameters:

mapp: Directory EIC of the table.

mn: Controls loading of arrays mbar and mkey as follows:

1 or 2 will load the name value into array mbar.
2 will also load the key value only into array mkey.
3 or 4 will load the mbar array with the key value (in brackets) followed by the name value. This construction will appear as: [key-value] name-value.
4 will also load the key value only into array mkey.

mbar: An existing memory array sent by reference into which the name values or "[key-value] name-value" constructions for the table entries are to be loaded.

mkey: An existing memory array sent by reference into which the key values are to be loaded. The mkey array must be present in the parameter list only if the mn parameter value is either 2 or 4. The key values returned in the mkey array correspond in array sequence to the values loaded into the mbar array.

NOTE: Both arrays mbar and mkey will be redimensioned to the number of table entries loaded by this routine.

Returned value: Returns the numeric number of entries loaded. Returns a 0 value if no entries are loaded.

Operation: If the memory variable "mlgth" is available to this routine, it will be set to the maximum length of any of the name values or "[key-value] name-value" constructions that were loaded into array mbar. The user must assure that there is sufficient memory space available to load the applicable arrays.

This routine loads only the table name and key values. The memo field contents associated with TM, TC, and TN tables will not be loaded by this routine.

Environment:

TALK must be OFF.
DELETED must be ON.
Current Database selection at call is restored at exit.
No user interaction.
Calls: None.
Databases: May use ZTAB or ZTAM DBF files.

Function ZULST (@mbar, mnum, mlgth)

Purpose: To display the contents of a memory array as a list.

Parameters:

mbar: The array to be displayed sent by reference.

mnum: The number of elements in array mbar to be displayed (1 through mnum). Array mbar can be larger than indicated by the mnum value.

mlgth: Optional parameter that specifies the maximum character length of any element in array mbar. If this parameter is not present, it will be calculated from the array contents.

Returned value: Returns logical True.

Operation: The routine will use a display window beginning at row 3, column 3 and extending to the smaller of row (mnum+9) or row 23. The width of the text area of the window will be 21 characters if mlgth is less than 21 and 62 characters if mlgth is greater than 62. Otherwise, the width will be adjusted to accommodate the maximum element length as specified by mlgth.

Environment:

TALK must be OFF.
Windows used: WZ_lst.
Uses White on Red for color scheme.
Calls: None.
Databases: None.

Procedure ZPFLD

Purpose: To load the contents of a defined directory table from a DOS ASCII file.

Parameters: None.

Returned value: Public variable qf is set to 0 if processing of the DOS file was successful. Otherwise, qf will be set to 1 denoting either an abortive action or an unsuccessful load.

Operation: This utility must be activated from the "Table Edit Menu" of the CAMD Data Directory subsystem. The table code and associated control parameters must have been defined prior to using this option. The required DOS file format of table data is as follows:

For TA and TL tables: One record per table entry:
name-value.

For TB tables: One record per table entry:
key-value^name-value.

For TP tables: One record per table entry:
key-value^name-value^extra-identification-data.

For TM tables: First record of each table entry:
name-value^number-of-following-text-line-records.
Records 2 through (# lines + 1) for each entry:
Text-line.

For TN and TC tables: One record per matrix-cell-value. The records must be in row-significant order: (1,1), (1,2), (1,...), (2,1), (2,2), (2,...) etc. There must be a record present for every cell of the matrix.

NOTE: For TC and TN tables, this utility can be used to completely replace the contents of an existing matrix if desired. For all other table types or editing of the individual entries of TC and TN tables, the normal table entry edit process should be used.

Any TB or TP load record having a duplicate key value will be ignored. A duplicated name value found during the loading process for any table type will be presented in an alert window where it either can be accepted or rejected at the user's discretion.

Once processing of the load file has begun, do not interrupt the process. A "process completed" message will be displayed along with a count of the number of records processed when the loading process is finished.

Environment:

TALK must be OFF.
DELETED must be ON.
Current Database selection at call must be the ZDIR.dbf file with pointer set to the directory code of the subject table.
All memory variables for the control parameters of the subject table must be available.
Windows used: W_pmt2 (predefined).
Uses color scheme 19 from set "DONS" in resource file ZRSRC.DBF.
Calls: Routines ZALERT, ZPD, ZUDUP, ZUFLDM, ZUFLDN and ZPTABF. Routine ZUFLDM is used to process files for TM table types. Routine ZUFLDN is used to process files for TC and TN table types.
Databases: May use any or all Z*.DBF files.

Function ZDP (mx, ms, mp, mp2)

Purpose: To extract and return the substring that is present between occurrences of a specific delimiter. This utility behaves in a manner identical to the MUMPS language $Piece function and utilizes the same parameter definitions and order.

Parameters:

mx: The source character string. If the length of mx is zero, the returned value will be null.

ms: The delimiter character string. ms must be at least one character, but can be any number of characters. Each occurrence of the delimiter identifies each substring of the source string as described in the following example where the asterisk (i.e., *) character is the delimiter string:

The source string (mx) is ABC*DEF*GHI

In this case, ABC is the first substring (piece), DEF is the second piece, and GHI is the third piece.

mp: A numeric value denoting the nth occurrence of the delimiter string ms within source string mx. mp represents the initial substring to be extracted. If the mp2 parameter is not present, it is assumed to have the same value as the mp parameter, and mp therefore represents the only substring to be extracted.

mp2: An optional parameter. mp2 needs to be present only when more than one consecutive delimited substring is to be extracted. A number value greater than or equal to the value of mp. If present, mp2 denotes the final substring to be extracted such that the returned string consists of all

characters following the mpth delimiter, up to but not including the mp2th delimiter or to the end of the source string is the mp2th delimiter is not present.

Returned value: Returns the character string mxp set to the extracted substring mp or mp through mp2 of the mx string.

Operation: If the delimiter does not occur within string mx and mp is 1, all characters within mx are returned. If mx is null or there are no characters following the mpth delimiter, null is returned for any values of mp or mp2. If the mpth substring is null and mp2 is not greater than mp, null is returned.

Environment:

TALK must be OFF.
Calls: None.
Databases: None.

Function ZUDUP (mapp, mtype, mkey, mtxt, mpars)

Purpose: To determine if the name and/or key value given is an exact duplicate of an existing table entry.

Parameters:

mapp: Directory EIC of the table.

mtype: Is the table type code TA, TB, TC, TM, TN, or TP. This function is not necessary for new entries to TL tables since all TL table entries are visible during entry or edit operations.

mkey: Applicable only to table types TB and TP, but must be present as a null value parameter for other table types. The mkey parameter is the given key value to be investigated which corresponds to the name text given by parameter mtxt.

mtxt: The name value to be investigated.

mpars: This parameter is the zparse control parameter of the subject table that determines the word parsing and conversion action regarding special characters found in the mtxt name value. If not present, the normal C value parsing action is assumed.

Returned value: Returns numeric value qf as follows:

qf = 0 Both mkey (if applicable) and mtxt are unique.
qf = 1 mkey is a duplicate key value.
qf = 2 mtxt is a duplicate name value.
qf = 3 Both mkey and mtxt are duplicate values.

Operation: This routine is used by the normal directory maintenance utility and is also called by the ZULAY (Learn-As-You-Go) utility to determine if a new table entry duplicates an existing entry.

Environment:

TALK must be OFF.
DELETED must be ON.
Current Database selection at call is restored at exit.
Calls: Routines Znamfix and Zpd.
Databases: May use any or all Z*.DBF files.

Function ZULAY (mapp, mdup, mtxt, miden, mxtra)

Purpose: Files a new table entry name value, and for TB and TP tables, a key value.

Parameters:

mapp: Directory EIC of the table.

mdup: A numeric parameter that governs checking for duplicate name or key values as follows:

0 = Check for duplicates and return mqf accordingly. The Zudup utility function will be used to perform this check.

1 = Do not check for duplicates. In this case, mqf can only be returned set to either 0 or 4.

mtxt: The name value for the new entry.

miden: The key value parameter. miden must be present only if the table type is TB or TP. This parameter is otherwise ignored and may have a null value if present in the parameter list. The trimmed length of miden must be greater than 0 and not greater than the length specified by the zkeylgth control parameter.

mxtra: This parameter must be present only for table types TB and TM. If present for other table types, it is ignored.

For TP tables, mxtra is a text string (up to 30 characters) containing the secondary person identification information in a format suitable for direct display during a selection process (e.g., "Sex: Male DOB: 9/23/45"). The mxtra information is always displayed following the name and key values of selection candidates.

For TM tables, mxtra is the memo text value as a character string. mxtra must be a memo field or the result of a @ row,col EDIT type of entry.

Returned value: Returns numeric value mqf as follows:

mqf = 0 Filed okay. If mqf > 0, not filed because:
mqf = 1 mkey is a duplicate key value.
mqf = 2 mtxt is a duplicate name value.
mqf = 3 Both mkey and mtxt are duplicate values.
mqf = 4 Invalid key (miden parameter) length.

Operation: The ZULAY routine can be used to add new entries to existing directory tables in a programmatic fashion. It is applicable to TA, TB, TM, and TP tables only. The TL, TN, and TC table types do not allow alteration except by means of the normal directory utility definition and edit process. If there is a possibility of duplication of a name (mtxt parameter) and such a duplication is to be allowed, the mdup parameter must be 1. The user can make this determination prior to calling ZULAY by first calling the ZUDUP utility function to identify duplications of either key or name values.

Environment:

TALK must be OFF.
DELETED must be ON.
Current Database selection at call is restored at exit.
Calls: Routines Zudup, Znamfix, Zdelsp, and Zpd.
Databases: May use any or all Z*.DBF files.

5.3 Data Directory Subsystem.

5.3.1 Data Directory Utilities Software. The Data Directory Utilities software provides a means for creating all of the individual data elements and tables of information necessary to support an operational application-specific module. This software provides the identification, definition, and editing utilities for the creation of a variety of data elements and associated parameter information necessary to control their normal use within an application module or subsystem utility. A data element can be a numeric value, a character string, a date, a time, a logical value, a free text document, or one of a variety of table configurations. The Data Directory also provides identity-definition for conceptual or abstract entities that may be native to certain programming styles such as the "Classes" and "Objects" are to object-oriented implementation schemes. Each directory element definition constitutes a "directory entry" and provides a unique global identity (code and name) to the individual element.

5.3.1.1 Data Element Composition. A data element as defined within the application support Data Directory establishes a specific informational or conceptual entity. Each data element maintained by the Data Directory minimally has the following components.

ELEMENT IDENTIFICATION CODE (EIC): The EIC consists of only uppercase letters A through Z and the digits 0 through 9. The first character of an EIC must be a letter. The remaining characters can be any sequence of letters or digits. All codes beginning with the letter Z (Z...) are reserved for use by the Data Directory Subsystem. The length of the EIC is fixed, typically at four or five characters. The selection of the EIC length is left to the discretion of the implementation authors. However, once established, its length cannot be altered. The EIC is the primary identifier for a particular data element and is unique within the system.

Typically, the author is allowed to choose an EIC for any data element being defined as long as it does not duplicate that of an existing data element and conforms to the above rules for EIC construction. If it is more advantageous for the utility software to automatically construct and assign the EICs, the algorithm used to construct the EIC should avoid using the vowels I, O, and U in character positions other than the first.

PRIMARY NAME: The Primary Name is the textual name given to the data element by the author. It is also a secondary means of identifying the data element in that directory authors can use this name to identify a directory data element entry for display or editing actions. The name can consist of up to 64 alphanumeric characters including both uppercase and lowercase letters and all special characters except the "^" and ";" characters. The Primary Name has both an "External" and "Internal" format. The External name is retained exactly as it was input by the author. The Internal name is the External name configured such that all lowercase letters are converted to uppercase, all special characters are converted to spaces, and all multiple spaces found between name terms are converted to a single space. The Internal version of the name is used only by the directory name retrieval utility routines for name search actions.

DATA ELEMENT TYPE: A data element must be defined as any one of the following general data types. The design of the application support utilities assumes that all filed informational elements belong to one of the types of data described below. The data type can also be thought of as the assigned "class" of data. All data elements are assigned a data type to facilitate understandability, ease of entry and manipulation, and to enhance the potential for storage optimization. The letter code indicated here for each type is provided as an identification

code for the particular type for discussion purposes only. The application support authors can employ any appropriate coding scheme for the data type code.

Type C: A character string value. Type C elements are a text value having a maximum length of 254 characters. All alphanumeric and special characters except the "^" can appear in the value.

Type D: A calendar date value. The entry and storage formats of date values should be selected to conform to the general application usage requirements and, if possible, to existing date processing functions within the processing language.

Type E: A time value. The entry formats and storage format of time values should be selected to conform to the general application usage requirements and, if possible, to existing time processing functions within the processing language.

Type I: An identity value. This data type is used to provide a unique identifier and name for use as a heading or title text, or to give identity to any general real or conceptual entity such as a topic, subtopic, object, class, etc. The only values associated with this element type are its EIC, data type, and Primary Name. The Primary Name can be up to 64 characters in length. No other parameters are related to this element type. Application support data-entry utilities will not process (allow input) of this data type.

Type L: A logical value that can only be interpreted as "true" or "false."

Type M: A multiline free text that can be processed by an appropriate word-processing or line-editing utility. This data type can also be referred to as a document, notation, comment, or memo data type. The syntactic configuration and content of values for type M data are restricted only by the editing utilities employed to process this data type. Other parameters are employed to further describe the "text" configuration of this data type.

Type N: A numeric value. Other parameters are used to further describe the configuration of the numeric data type.

Type R: A reference value. This element type is always constructed internally. Its content and configuration are unknown to and unrestricted by the directory system. It is considered within the application support utilities to be a string (character) value. Application support data-entry or

display utilities do not process this data type. It is a "computed" value and is typically used as an internal pointer, locator, or as an indirect reference to a program, function, or graphic image.

Type T*: The "T" denotes a table type data element. The second character "*" is used to denote the letter code of any one of the specific table types described below. In all table types where a name (or text) value is involved and for which retrieval and selection of individual table members is done by means of a name criterion, the names values (primary and secondary) follow the same construction and length rules set forth above for the Primary Name of the directory entries.

TA: A table of multiterm name/text values, each of which can be up to an author-specified maximum length. A parameter value (ROTATE) for this table specifies whether or not the name/text entries are to be stored for retrieval purposes. If not, selection can only be accomplished by means of a "pop up" list as described for table type TL, or directly by the relative-location index value assigned to each entry. This feature allows the table to be used as a storage file for text-string values where name/text retrieval is not required. Normal retrieval and selection of a table member is by partial match of one or more name terms. Table size is unlimited. Normal use of this table type is storage and retrieval from a relatively large group of subject names or text-string values.

TB: A table of multiterm "primary" name values with an associated "key" value, and any number of multiterm "secondary" name values. The length of each primary or secondary name can be up to a maximum specified by the author. The associated key must be from 1 to 9 characters in length and can be any alphanumeric configuration. The key values used for members of this table type are governed by the "KEYS" parameter. Each key must be unique within the table. Retrieval and selection of a table entry is by (1) exact match of a key value, or (2) partial match on one or more name terms. Table size is unlimited. Normal use of this table type is storage and retrieval of a relatively large group of keyed (associated codes) name values.

TL: A "list" or "menu" type of selection table where all selection options (table members) are always displayed as a selection menu. The entries to this table type are limited to 30 characters. Selection is by (1) selection from a "pop up" list, or (2) marking a selection by other employed conventions which indicates

a selection action. Selections from this table type can be limited to one, or any number of selections can be allowed. Although the size of this table is unlimited, its normal fully displayable selection list dictates that it be used only for a relatively small number of entries since all options must be held in a memory array during selection. The order of option presentation can be manipulated to conform to the author's design. The TL table type is very similar to type TA and TB types, but utilizes a completely different method of selection.

TM: A table where each member is a multiline free text entry as described for data type M above. Each member is given a name consisting of up to 30 characters which then can be used for retrieval and selection of the individual text entry. Retrieval action is as described for type TA table member names.

TP: A table of personnel names with associated identification codes. Name entry format is as follows: LAST, FIRST INITIAL(S) with the space following the comma optional to entry personnel. The "LAST" and "FIRST" name areas can contain more than one name term with terms separated by one space. At least one character must be present for the FIRST name, and the presence of initials is optional. Entries in this table are optionally cross-referenced by one or more "identification code" values. These codes are typically the Social Security Number and/or an employee identification code. Each such code provided must have values that are unique among all codes values employed. Retrieval and selection of table members is by (1) exact match on any identification code, (2) partial match of the LAST name term, partial match of LAST and partial match of FIRST name terms, or (3) partial match of the LAST name term with an "*" denoting acceptability of any first name. Maximum total length of an entry name is author-specified. The table size is unlimited. Normal use of this table type is for storage and retrieval of personnel identities. Each entry of a TB table is actually located by a relative-location index key. All identification code and member name terms reference this key, which can never change value and will always reference the particular individual regardless of changes to associated identification codes or member names.

TN and **TC:** The TN and TC table types both represent either a 1-dimensional or 2-dimensional array of values (elements). The elements of a TC table type are character type (type C) values. The elements of a TN

table type are numeric (type N) values. All elements of a table are the same data type. A 2-dimensional array is referred to as a "matrix" table. A 1-dimensional array can be referred to as either a "matrix" or a "vector." Either table is limited to a maximum of 3,600 individual elements. The extent of the array is defined by "number-of-rows" for a 1-dimensional array (vector) and also by "number-of-columns" for a 2-dimensional array (matrix). Each element is individually referenced by either one or two integer "subscript" values, relative to the dimensional aspect of the table. Subscript values always begin at 1. Elements of a 2-dimensional matrix are normally referenced by a "row,column" set of subscripts. However, they can also be referenced by a single subscript value that denotes the result of the following formula: Subscript = ((row# -1) * (number-of-columns)) + column#. The above formula dictates that 2-dimensional matrix elements are stored in "row-significant" order as follows: (1,1), (1,2), ..., (1,n), (2,1), (2,2), etc. Also, relative to a 2-dimensional table, the "row" elements (1,1), (1,2), ..., (1,n) can be referred to as a "row-vector", and the "column" elements (1,1), (2,1), ..., (n,1) can be referred to as a "column-vector." Entry utilities for input of matrix values can be controlled by parameters to read one element, a row-vector or column-vector, or the entire array in either row or column order. A 2-dimensional table can be given "title" names (up to 30 characters) for the row- and column-vectors. The primary name serves as the title of a 1-dimensional table. The row and/or column vectors can be named via the MATREF parameter.

5.3.1.2 Data Element Control Parameters. In addition to the EIC, Primary Name, and data type code, each directory entry (except type I) must have a set of parameter values that are used to govern the data entry or selection processes and other operations involving the data element. There are several general parameters that are common to all data types. Other parameters are specific to particular data types. All parameters noted for the definition of any particular data type must be solicited and maintained in the directory regardless of the value given to a parameter.

For the following descriptions, each parameter has been given a name. These names are arbitrary and used here to uniquely identify each parameter for discussion purposes. These names will be used throughout all application support utilities documentation whenever there is a reference to a data element control parameter. If desired and appropriate, application support authors can adopt these names exactly as they appear

below as actual field or variable names for the respective parameter values. The directory code, Primary Name, and data type are also considered to be control parameters and are listed here. The term "null" that is used means that the subject value is defined but empty (contains no characters).

Note: Data element type I is not mentioned in any of the following parameter descriptions since no parameters are applicable to type I elements other than the EIC, NAME, and TYPE.

DELT: Applicable to all types except R. Denotes whether or not the existing value of a filed data element being used as a default value can be set to a null value (deleted) during an edit action. Values for DELT are either 0 to allow deletion of an existing response (default response) value, or 1 to not allow an existing response to be set to a null value. The default value is 0. Application support entry utilities allow this setting to be overridden at an input action.

DIM: Applicable to TC and TN table types. For TC and TN tables, DIM denotes the array dimensions for either a 1- or 2-dimensional matrix in the form: #rows or #rows,#columns respectively.

DISP: Applicable to TA, TB, TP, and TM table types. This parameter denotes if a list of all table entries is to be displayed prior to prompting for a selection. The values for DISPTBL are 0 to not display the table entries before prompting, or 1 to do so. If 1 is used, the prompt value PMT is used as a table display heading as described for TA table types in the PMT parameter description. The default value for DISP is 0. Application support entry utilities allow this setting to be overridden at an input action.

DKEY: Applicable to TB and TP table types. Denotes whether or not to include the "key" values during a display of the table entries or a list of candidates for selection actions. If DKEY is "OFF," only the table name/text values are displayed. If DKEY is "ON," the key value, in bracket delimiters, is concatenated with the entry or candidate name/text value for presentation.

EDIT: Applicable to C, D, E, N, and all T* types. Provides an executable reference to a response editing or validating process or function. Null, the default value, denotes that no reference is present. If present, EDIT is used as the "VALIDATE" clause for entry or edit actions and must conform to the FoxPro 2.0 requirements for this clause.

EDITOR: Applicable to M and TM types. Denotes the type of orientation within the body of text. The body of text can be either a word-oriented text, where words that extend beyond a specified right margin are "wrapped" to the next line, or a line-oriented text where each line is a logical entity and no word-

wrapping or other line-content manipulation is done. The values for EDITOR are 0 for word-oriented text, or 1 for line-oriented text. The default value is 0.

EIC: Applicable to all data types. The 4- or 5-character alphanumeric code that uniquely identifies a data element entry in the Data Directory.

FILECD: Applicable to all types except R. Other utilities within the application support utilities maintain a key-coded table where each entry is a reference to a particular database file or file structure. These references are used to control and locate the exact field, position, and configuration for storage of each data element within the application database. The key that uniquely identifies each file-reference table entry is a 3-letter code. The parameter FILECD can be assigned any existing key for this table as a "default" file-code. This parameter can be superseded or dynamically replaced by a number of application support operations and represents the actual data element filing placement only in the complete absence of higher order filing or retrieval criterion. The field can be null. A null value does not indicate that the data element is never filed, although that may be the case since any directory data element can be input and used as only a temporary value. The default value for FILECD is null. Also, any data element can be filed in more than one place within the application database.

FRAC: For type N and TN elements, FRAC is the fractional component for elements that represent real-number values. The fractional component indicates the placement of the decimal for values of the element.

HELP: Applicable to C, D, E, N, and all T* types. Provides either an executable reference to a help display function or, if HELP is the EIC of a directory code, the "detail" (memo) field for the EIC code (Topic) contained in the application support "HELP" file will be displayed. Null, the default value, denotes that no reference is present.

HIST: Applicable to all types except R. Denotes the requirement that a historical recording must be made when the current value (or set of values) for this data element is altered. Values for HIST are 0 to not perform historical recording, or 1 to require historical recording.

KEYLGTH: Applicable only to TB and TP table types. Denotes the maximum allowed character or digit length for the key values. A value for KEYLENGTH that contains a decimal point denotes the position of a decimal in the key values. No actual key value can exceed this given length.

KEYS: Applicable only to TB and TP table types. Denotes the source of the key value(s) for new table entries as follows:

1: Not keyed (default for all other table types).
2: Keys are externally provided numeric values.
3: Keys are externally provided single-letter codes A through Z. The letter value assignment does not have to follow the letter collating sequence.
4: Keys are externally provided alphanumeric codes.

The term "externally provide" as used above means that the value for keys is not automatically computed within the table entry utility routine. The key value for a new entry is either supplied as a parameter or is prompted for. Note that option 3 will automatically limit the table to a maximum of 26 entries.

LAYGO: Applicable only to TA, TB, TM, and TP types. Denotes whether or not the end-user has the capability to add a new entry to the data element table during a search-and-select action. This is a different action than that used by the application support utilities and application authors to define or edit table entries in that a LAYGO (Learn-As-You-GO) action is invoked by an unsuccessful attempt to find an existing entry that matches a name value entered by the application user. Application support utilities do not directly support LAYGO actions. They only provide application authors with the appropriate entry and filing functions that can be used to implement LAYGO actions. Values for LAYGO are 0 to not allow Learn-As-You-GO actions, or 1 to allow this action.

LGTH: For types C, D, E, N, R, and T*. Author-specified maximum character length allowed for the type C, D, E, M, N, and R type value or TA, TB, TL, TP, TM, TN, and TC table entry name value. The maximum length that can be specified for C, D, E, N, and R is as follows: C=254, D=16, E=12, N=20, R=no specified limit. For table types TA, TB, TL, and TP, LGTH represents the maximum length allowed for the name/text/option value of the table members and can be given any value up to 64 characters. For table type TN, LGTH can be up to 20; for type TC, LGTH can be up to 64. For type TM, LGTH represents the maximum number of characters that can be used for the "name" given each memo entry for retrieval and selection purposes (max=64). The LGTH value for the L (logical) data type is automatically defined as 1. The default value for this parameter will be the maximum number of characters allowed for the respective data type as noted above.

Note: Table type TB or TP names should not allow a name (selection option) length that when joined with the corresponding key and key-name separation characters would extend beyond the right screen margin and cause "wrapping" or truncation of the selection text. This parameter is a mandatory entry for all

element types except R. It is used by the data-entry utilities to validate the input response length.

LIST: Applicable to TA, TB, TP, and TM table types. Denotes whether or not a list of all table entries can be displayed during an entry action upon request (usually by entry of a "?" character) from the end-user. The values for LIST are 0 to not provide the list, or 1 to provide it. The typical action to provide the list is the following. Upon recognition of a "Help" request from the user or the failure to find at least one candidate selection after a search has been conducted, the user is asked if a list of all entries is wanted or not. This prompt should be given only if the display of the list would exceed the available screen area. The default value for LIST is 0.

MATIP: Applicable to TN and TC table types. Indicates the method to be used to enter values into the vector or matrix table as follows:

0: No entry; all values are preloaded.
1: Enter individual elements by row.
2: Enter individual elements by column.
3: Enter row-vector elements as a multivalue string with space separators.
4: Enter column-vector elements as a multivalue string with space separators.
5: Enter all elements in row-significant order.

MATREF: Applicable to TN and TC table types. If this parameter is not null, it must contain either one or two EIC identifiers of existing TA or TL tables whose name entries are to be used as row and/or column labels. The definition formats for the parameter are as follows:

Format for a vector table: One EIC only.

Format for a matrix table: EIC-for-rows,EIC-for-columns where either EIC-for-xxx reference can be null denoting that the row or column elements are not labeled. At least one reference must be present, either before or following the comma separator.

Note: The number of existing entries of a referenced table must exactly match the respective number of elements assigned for the vector or row and column dimensions of the table array. Also, the order of the table entries must match the sequential order of the respective subscript for the corresponding array entries.

MAXKEY: Applicable to TB and TP table types. Denotes the maximum number of key values that are allowed to be entered as a pointer to the name/text member of the table. The default value of MAXKEY is 1. If MAXKEY is given a value greater than 1 or if

the LAYGO parameter allows Learn-As-You-GO action, further parameter information concerning the source of the key values will be prompted for. Namely, each key will necessarily have to be defined and identified as a type C or N data element so that the LAYGO process can prompt for the key values to be associated with the name or text values of new table entries.

MULT: Applicable only to the TL table type. The parameter indicates whether or not a user can make multiple selections from the list of table entries. The default value for MULT is 0 to allow only one selection. A value greater than 0 denotes the maximum number of selections that can be made at a single prompting action. Multiselection typically is implemented by either allowing entry of a string of key values (keys are separated by either a space or comma), or by a host-language supported means by which the user can "mark" desired selections.

NAME: Applicable to all data types. The Primary Name given to the data element.

NULL: Applicable to all types except R. Denotes if the data element, in general, is allowed to have a "null" value. This parameter can be overridden during an input action. Values for NULL are 0 to allow a null response, or 1 to not allow a null response. The default value is 0.

NUMTYP: Applicable to N and TN types. Denotes the general type of numeric value as follows:

1: Integer only.
2: Integer representing a dollar amount (no fractional component).
3: A real-number value (+, -, and fractional component allowed).
4: A dollar amount with a 2-place fractional component.

OKAY: Applicable to TA, TB, TM, and TP table types. This parameter denotes whether or not a verification prompt is to be issued when only a single candidate results from a search action. The prompt is to verify that it is in fact the intended entry. The typical prompt is for a yes or no response to "Okay?" or "Is this the one?". This parameter is applicable only if the DISP parameter is "OFF" (no table display). An "OFF" value for OKAY, the default, denotes that no verification prompt is to be given. An "ON" value invokes the verification prompt.

PARSE: Applicable to TA, TB, and TM table types. This parameter controls the conversion of the name/text value of a table entry from the external form to the internal form. Only the internal form is used to store the name terms for name-oriented search and retrieval actions. Many names contain special characters such as dashes and parentheses, making retrieval on these terms difficult

for a user that may not be aware that they were present. For example, the term "(APPLE)" cannot be retrieved by giving "APPLE" as a search criterion. The PARSE parameter can be assigned one or more of the edit actions listed below. Each action that is set to "ON" activates the corresponding edit action for converting external name values to internal values. The edit actions are executed in the right-to-left order of their appearance. After all edit actions indicated by the PARSE parameter are done, all multiple spaces within the internal name version are condensed to a single space, and all lowercase letters are converted to uppercase.

Remove all special characters; replace with spaces.
Remove all dash characters; replace with spaces.
Remove all parentheses; replace with spaces.

A null value for PARSE denotes that none of the above edit actions are to be performed. The default value for PARSE is the third action listed above.

PIC: Applicable to all table types except R. If present, this parameter is used as the "PICTURE" clause for entry or edit of the element and must conform to the FoxPro 2.0 conventions for construction and use of the PICTURE clause. For table type TL elements, PIC can contain the PICTURE codes *C, @&, @^, @*, and @*R to denote that the table options are to assume the entry behavior of "check box," "list," "pop up," "push button," or "radio button," respectively.

PMT: Applicable to all types except R. This parameter provides, either directly or indirectly, the general prompt text to be used during an entry action for the data element. If PMT is null, the Primary Name is used as the prompt. If PMT is in the format DIRCODE:key and the code denoted by DIRCODE is a TA, TB, TL, or TM table having a member identified by the given key value, then the indicated text, name, or memo entry is used as the prompt text. The value of PMT otherwise can be a "literal" value which is the prompt text. The value of PMT assigned here can be overridden at any entry action. The default value of PMT is null.

Note: For TL table entry actions, the PMT prompt value is used as a menu heading, displayed before the selection list rather than after, and a "Section>" or similar term is used to tell the user that selection can begin, if any such term is required.

RNG: Applicable only to N and TN types. Provides a "low" and "high" value numeric range allowed for the element value(s). The entry format for this parameter is a response to both a "low" and "high" value prompt. The minus sign (-) can appear on either the low-value only or on both values. Either value can contain a decimal point. Null, the default value, denotes that no value range is present. If this parameter is not null, it is

automatically used to validate the responses for the element during entry.

ROTATE: Applicable only to table type TA, TB, and TM. This parameter is used to denote whether or not the table entry utility routine is to parse and store the name/text value of the entry in the "name term" directory file for name-retrieval operation.

TYPE: Applicable to all data types. The code (C, D, E, I, L, M, N, R, or T*) specifies the basic configuration and usage type of a data element.

UNITS: Applicable to N and TN data types. It specifies the unit or scale that is associated with the numeric value(s) and is a character-type field. Null denotes no direct unit association.

XDATA: Applicable only to table type TP. This parameter specifies one or more existing directory codes (EIC) of elements that are to be used as displayed information for selection candidates resulting from a search action performed on the TP table. The elements referenced by this parameter typically are sex and date of birth. The display of this information in addition to the key and name for selection candidates is to provide differentiating information to the user in the case of duplicate candidates. The parameter can be null. If it is used, the elements referenced must already be defined as directory elements and must be type C, D, N, or TL. Parameter format is EIC,EIC,...,EIC.

Within all Data Directory Subsystem programs, the names for all of the above described parameters are prefaced by the letter Z when used as a memory variable (e.g., EIC becomes ZEIC, NULL becomes ZNULL, and so on).

5.3.2 Directory Utility Routines. To support authoring of data elements in the directory, the application support utilities must provide the capability to create, edit, and display the directory entries. Notice that there is no mention of a "delete" operation because there should not be a utility process that has the ability to delete an existing directory entry. This feature insures that at no time can a data element be deleted from the directory after it has been defined for an application.

The structures of the Directory files are not dictated in these specifications since they depend on the host software and database implementation methods used by the application authors. However, it is expected that a single file, indexed on the EIC, can serve to contain the name and parameter set for all directory entries. Several file structures will be required to implement the storage and retrieval operations pertaining to the table types (T*) of data elements.

5.3.2.1 Create/Edit Utility. The utility routine for creation or editing of directory entries can be essentially the same routine. The creation operation of the create/edit utility routine must assure that the EIC given for the new entry is not already in use. As noted in the description of the EIC, the EIC constructed from all Z letters is reserved. This restriction is because the codes and names of the directory entries themselves will become the table members of a type TB data element having the ZZ... code as the EIC and the name "DIRECTORY ENTRIES." Therefore, retrieval and selection of the directory entries for the purpose of edit or display can be accomplished using the same retrieval utility that is used to search and select from any type TB data element table. Any retrieval action using a new EIC as a search criterion that fails will provide assurance that the EIC is not in use. Likewise, for editing or display actions, existing entries can be identified using either the EIC or by name search. Again, notice that there is no mention of a "delete" operation because there should not be a utility process that has the ability to delete an existing directory entry. This feature serve to insure that at no time can a data element be deleted from the directory after it has been defined for an application.

5.3.2.2 Display Utility. The Display utility must initially offer a review of directory entries by individual selection, by selection of entry groups having one or more beginning characters of the EIC in common, and by specifying that all entries are to be included in the review. If the selected option is other than by an individual entry, a secondary option should allow the directory author to specify the entry output order to be in either EIC order or in alpha order of the Primary Names.

The content of the displayed output should be selected as follows for any of the above mentioned entry selection options:

Option 1: Include EIC, Primary Name, and data type.
Option 2: Include control parameter settings.
Option 3: Include list of table/matrix entries for T* data types.

The option selection process must allow any combination of individual options or all options to be selected. Also, the author should be allowed to direct the output to either the terminal or the printer device.

The output for control parameter settings should include the name and value given to all parameters applicable to the data type, noting null values. Output for the TA, TB, TL, TM, and TP table entries should include the relative index numbers or key(s) as well as the name/text values. For TN table entries, the standard display will be to horizontally display vector and matrix row values wrapping to as many lines as necessary to display all entries in the vector or row. For TC tables, one

entry per display line will be used. The element display for 2-dimensional arrays will be by row-significant order.

5.3.2.3 Table Entry Utilities. The utility routines that service T* table type data elements provide the capability to add new entries, edit existing entries, and display or print the existing entries. For new entries, both the prompting and filing utilities should operate as independent functions to facilitate a variety of table loading methods and flexible use by application authors. In general, authors should have the option of either using the table entry key/name prompting and filing facilities to solicit a new entry or "pass" the information for a new entry to just the filing operation. In the latter case, it must be the application author's responsibility to validate the data passed to the table filing routine. Entry of TC and TN table elements must follow the process indicated by the MATIP parameter.

If the application authors impose a length restriction for table name values or utilize the FoxPro memo type of field to store table entry names, then a single utility file can be used to store the name value for all TA, TB, TL, TP, and possibly TM table entries. Type TN and TC tables will necessarily require a dedicated file for each table type.

5.3.2.4 Retrieval from TA and TB Tables. Both type TA and TB tables allow retrieval and selection of entries by multiterm and partial term search criteria. This capability means that the search criterion can consist of one or more terms, each of which must match the beginning of any term within a table entry name. This type of search affords the user (and author) a very robust retrieval capacity that allows a search to range from very finite to very general depending upon the completeness of the given search criterion. The typical method for implementing this capability is by use of a "Term" directory, sometimes referred to as a "Rotary" directory. The Term directory is simply a file containing each "converted" term of the original name or text value. The Term directory uses the name term field as the file's primary index.

The term "converted" means that an internal version of the original name is prepared according to the conversion processes indicated by the "PARSE" parameter. After all conversion actions indicated by the PARSE parameter are done, all multiple spaces within the internal name version are condensed to a single space and all lowercase letters are converted to uppercase. The filing process parses the internal name on each term having a space separator and files the term as an index value of the Term directory along with other fields that provide the associated EIC of the Data Directory entry and the key (or pointer) to the external version of the table name. This process facilitates a rapid and very restrained "not-exact" search to be done in the Term directory to identify matching terms and construct a list of

selection candidates when the terms within more the one table entry name match all of a given search criterion. The given search term or terms must undergo the same external-to-internal conversion process.

For TB table types, if the search criterion contains only a single term, the search function must first attempt to exactly match an entry key value. If this match is successful, the target entry has been found. If this search fails, the search function assumes that the search criterion is a name term rather than a key and commence the "nonexact" match of name terms.

The utility that implements functions to search and select table entries must have these two functions as independent processes. A "search" function will invoke a search and return a list of candidates found that satisfies the search criterion. The "selection" function simply allows selection from the list of candidates. This feature provides a means of searching for an existing name (or key) that is duplicated by a new entry and also isolates functions that interface directly with the users or application authors. The only table that allows duplication of a name value is table type TP.

5.3.2.5 Retrieval from Other Table Types. Retrieval from TP type tables is essentially the same process as described for the TB table type. The differences are the following. The search criterion entry follows a specific person-name format. Duplicate candidates are possible and allowed. The retrieval format is discussed under the description given for data type TP. Retrieval for TL table entries will be done only by the particular menu-selection function utility implemented for this purpose in the user interface utilities. Retrieval for TM entries will be by match of key values.

5.3.3 Data Element Input/Selection Testing. It is highly desirable that the author of a directory element have the capability to test the appearance and accuracy of an input or selection action for a newly defined directory entry as an option within the directory maintenance utilities. This test process should, as much as possible, use the same application support data element entry or retrieval routines that are available for application development use. Directory authors must assure that all referenced processes are in place before invoking test functions.

5.3.4 Additional Run-Time Parameters. The definition of data element parameters presented here describes only those parameters solicited and stored with Data Directory entries. There are other "run-time" parameters that can be used to further direct and control the behavior of entry and selection actions for data elements. The directory parameters that can be overridden at run-time are DELT, DISP, FILECD, HELP, LIST, NULL, and PMT.

5.3.5 Data Element Entry and Selection Utilities. The application support utilities provide utility routines that execute either entry or selection actions, as appropriate, for all data types except I, R, TN, and TC. There is a defined utility function for each data type that utilizes the directory parameters to control a single input, edit, or retrieval action for the given data element and returns the result to the calling operation along with a status indicator as described below. The implementation details of these utilities are largely dependent upon the behavioral scheme adopted for general interfacing of application interactions with the end-users. It is appropriate to specify only the general functional requirements here.

For entry actions of single-value data type elements (C, D, E, L, M, and N), the utilities support the following four actions. Note the corresponding value of a status "STAT" indicator:

1. A new value or null value is returned where no previous value (default value) was initially provided. STAT = 1.
2. A given (default) value is altered. STAT = 2.
3. Deletion of a given (default) value. This means that a non-null default was set to a null value by the user. STAT = 3.
4. The original (or null) value was unchanged. STAT = 0.

For table selection (retrieval) actions, only the following two outcomes are supported:

1. A successful selection was made. STAT = 0.
2. One or more candidates were found, but no selection was made. STAT = 1.
3. No candidates were found. STAT = 2.

The default and all result values, including returned key values and status indicators, are provided by means of standardized memory variables of the appropriate data type that are dedicated and reserved for this purpose. Other user requests that are allowed during entry or selection actions, such as "previous prompt", "previous screen", "next screen", etc. are to be implemented as required. User requests for "Help" information or table entry lists (when allowed) as well as validation actions should be processed within the utility.

5.3.6 Data Directory Usage and Aids. The Data Directory Subsystem is designed to reside in a "ZCAM" subdirectory of the primary FoxPro directory: Fox...\ZCAM where Fox... is the name of the main FoxPro directory. If the Data Directory Subsystem is to reside in a FoxPro subdirectory other than ZCAM, the SET DEFAULT TO... statement in routine ZPDIR must be altered to reflect the correct subdirectory.

The Data Directory Subsystem can be invoked from the FoxPro command window in either of the following two ways:

```
SET DEFAULT TO \Fox...\ZCAM
DO ZPDIR
```

or

```
DO \Fox...\ZCAM\ZPDIR
```

Three routines are available as an aid to testing or exercising the Data Directory Subsystem, the ZUTIP general code entry and selection routine, and browsing the directory files.

INIT DO INIT to initialize the directory files, environment, and memory variables. All nonwork directory files will be opened with alias names and indexes set:

Alias Names:

DIR ZDIR - The primary directory.
TAB ZTAB - The TA, TB, TL, and TP table entry file.
TAM ZTAM - The TM, TC, and TN table entry file.
DIC ZDIC - The system name lexicon file.

INIT does approximately the same initialization as the ZPDIR (Data Directory Subsystem invoke routine). If INIT has been run and you wish to invoke the Data Directory Subsystem (via DO ZPDIR), you can do so without using ZRESET.

ZRESET DO ZRESET to close all directory files, clear memory, and release all windows and menus. This reset must be done if the Data Directory Subsystem execution is interrupted by FoxPro because of error detection or any other reason.

ZTEST DO ZTEST to initialize a test window, parameter variables, and minimal environment to allow the use of the ZUTIP routine for testing entry and table selection behavior of defined data codes. ZTEST activates a typical I/P window and initializes the mdefault, mrkey, and mrdata parameters as PUBLIC variables. You then can reactivate the COMMAND window to manually enter calls to routine ZUTIP.

APPENDIX A. EXAMPLE OF QUESTIONS AND POSSIBLE ANSWERS USED BY THE BAYESIAN METHOD TO SUGGEST DIAGNOSES FOR CHEST PAIN

Questions about Patient	Allowable Responses to Questions
PATIENT CHARACTERISTICS	
Mood	0 = Normal, 1 = Anxious, 2 = Distressed, 3 = In Shock
Color	0 = Normal, 1 = Pale, 2 = Flushed, 3 = Cyanotic, 4 = Jaundiced (Check Palms, Conjunctiva)
Height	In nn.n inches
Weight	In nnn pounds (lbs.)
Build	0 = Thin, 1 = Average, 2 = Heavy, 3 = Obese
Smoking History	0 = Nonsmoker, 1 = Light Smoker (<1 pack/day), 2 = Moderate (1-2 packs/day), 3 = Heavy (>2 packs/day)
Smoker Status	0 = Nonsmoker, 1 = Smoker (automatically computed from Smoking History)
VITAL SIGNS	
Age	In nn (computed from Visit Date minus Date of Birth)
Temperature	In nnn.n degrees Fahrenheit
Pulse	In nnn beats per minute
Respiration	In nn respirations per minute
Systolic Blood Pressure	In nnn (mm. Hg.)
Diastolic Blood Pressure	In nnn (mm. Hg.)

Questions about Patient	Allowable Responses to Questions
CHEST PAIN DESCRIPTION	
Duration of Pain	In nn hours
Duration of Pain Ranges	0 = <1 hour, 1 = 1-2 hours, 3 = 2-4 hours, 4 = 4-12 hours, 5 = >12 hours
Onset of Pain	1 = Gradual, 2 = Sudden
Course of Pain	1 = Continuous, 2 = Intermittent
Site of Pain	1 = Substernal, 2 = Across, 3 = Left Side, 4 = Right Side, 5 = Epigastric, 6 = Other
Radiation of Pain	1 = Left Arm, 2 = Right Arm, 3 = Both Arms, 4 = Back, 5 = Shoulder, 6 = Neck, 7 = Jaw, 8 = Other, 0 = Does Not Radiate
Numbness (with/after Pain)	0 = No, 1 = Yes
Severity of Pain	0 = Moderate, 1 = Severe
Progress of Pain	0 = Better, 1 = No Change, 2 = Worse
Aggravating Factors	1 = Movement, 2 = Sitting, 3 = Cough, 4 = Breathing, 5 = Other, 0 = No Aggravation
Relieving Factors	1 = Nitro, 2 = Rest, 3 = Walking, 4 = Other, 5 = None
Type of Pain (Description)	1 = Tight, 2 = Sharp, 3 = Gripping, 4 = Burning, 5 = Dull, 6 = Stabbing, 7 = Nagging, 8 = Aching, 9 = Heavy, 10 = Crushing, 11 = Pressing

Questions about Patient	Allowable Responses to Questions
OTHER CHEST PAIN SYMPTOMS	
Shortness of Breath (Dyspnea)	0 = No, 1 = This Illness, 2 = Habitual
Paroxysmal Nocturnal Dyspnea	0 = Absent, 1 = Present
Cough	0 = No, 1 = This Illness, 2 = Chronic
Nausea	0 = No, 1 = Yes
Vomiting	0 = No, 1 = Yes
Appetite	0 = Normal, 1 = Increased, 2 = Decreased
Bowels	0 = Normal, 1 = Constipated, 2 = Diarrhea
Sputum	0 = Absent, 1 = Present
Orthopnea	0 = Absent, 1 = Present
Esophageal Reflux Symptoms	0 = Absent, 1 = Present

Questions about Patient	Allowable Responses to Questions
PATIENT HISTORY	
Previous Chest Pain	0 = No, 1 = Yes
Previous Cardio-Pulmonary	0 = No, 1 = Yes
Previous History of	1 = Myocardial Infarction, 2 = Angina, 3 = Bronchitis, 4 = Hypertension, 5 = Diabetes, 0 = No History of These Diseases
Previous Major Surgery	0 = No, 1 = Yes
Smoker	0 = No, 1 = Yes
PHYSICAL EXAMINATION	
Sweating	0 = No, 1 = Yes
Shivering	0 = No, 1 = Yes
Cold or Clammy	0 = No, 1 = Yes
Jugular Venous Pulse	0 = Normal, 1 = Raised
Respiratory Movement	0 = Normal, 1 = Abnormal
Heart Sounds	0 = Normal, 1 = Abnormal, 2 = PVC's (Enter _____) 3 = S3/S4 (Enter _____
Percussion	0 = Normal, 1 = Dull, 2 = Hyper-Resonant
Chest Sounds	0 = Normal, 1 = Rhonchi, 2 = Rales, 3 = Decreased
Edema (Swelling)	0 = Absent (None), 1 = Ankles, 2 = Other
Calf Tenderness	0 = No, 1 = Yes
Chest Wall Tenderness	0 = No, 1 = Yes

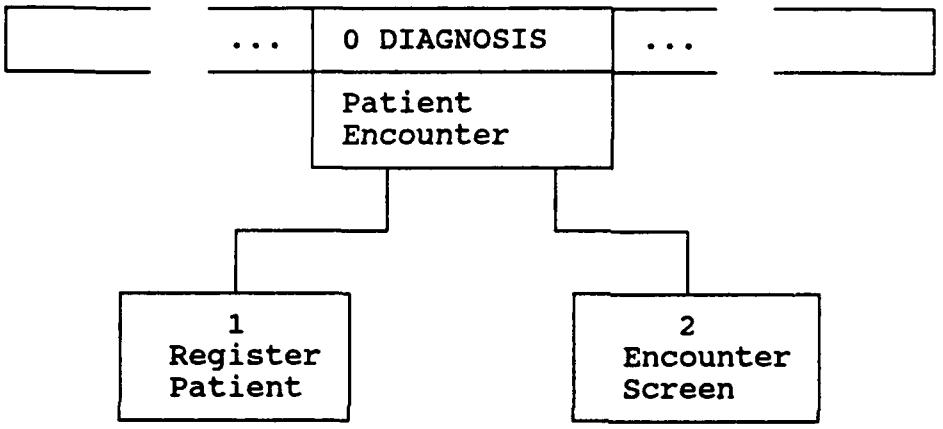
Questions about Patient	Allowable Responses to Questions
LABORATORY/ECG RESULTS	
SGOT	In nnn (IU/L)
SGOT Ranges	1 = <25, 2 = 25-50, 3 = 51-100, 4 = 101-200, 5 = >200
ECG Results	1 = ST Elevation, 2 = ST Depression, 3 = T Depression, 4 = Q Waves, 5 = Arrythmia, 0 = Within Normal Limits

APPENDIX B. CAMD EXPERT SYSTEM MAIN MENU AND SCREENS

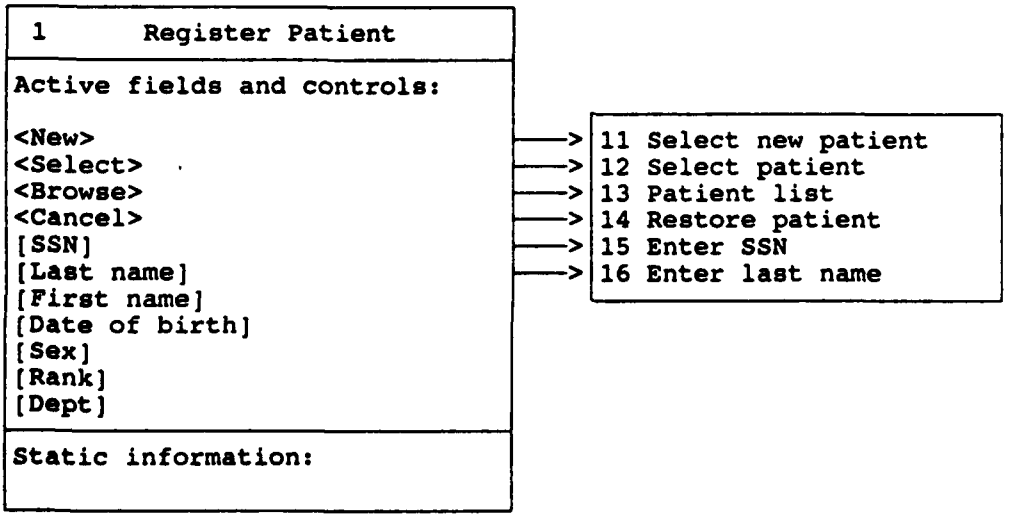
INTRODUCTION

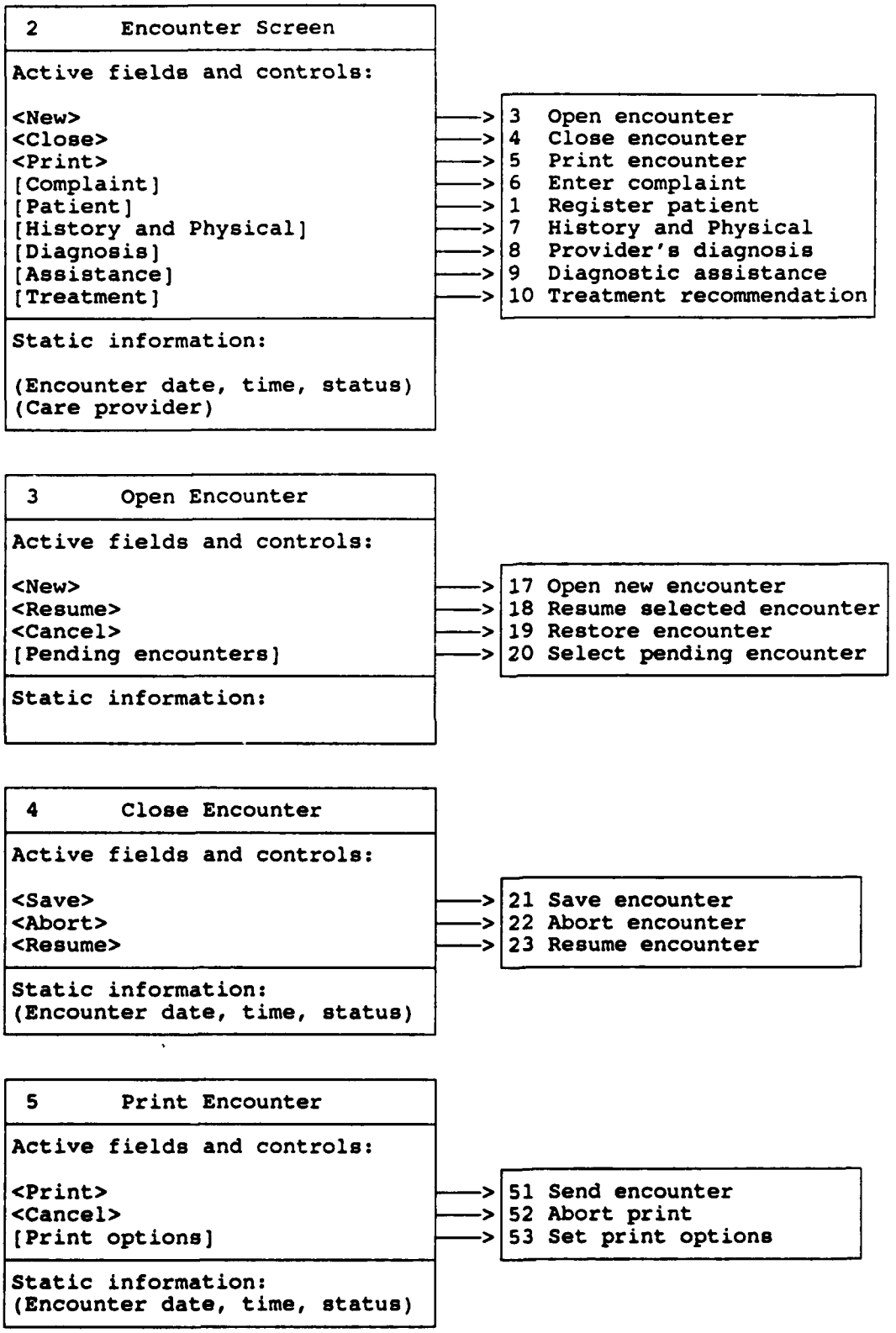
This document describes the CAMD expert system main menu and screens. This documentation is to serve as the main user interface specification.

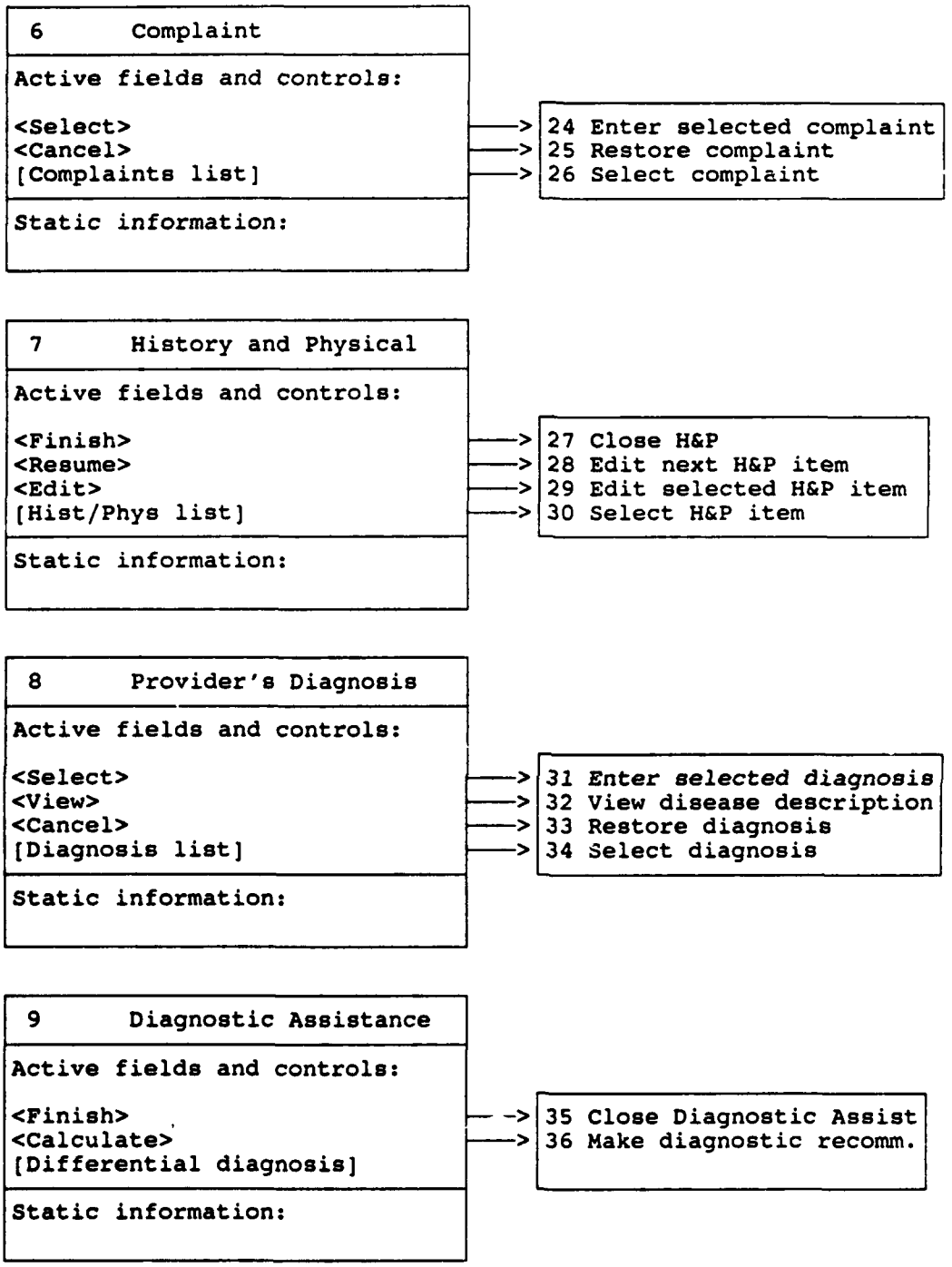
CAMD MAIN MENU

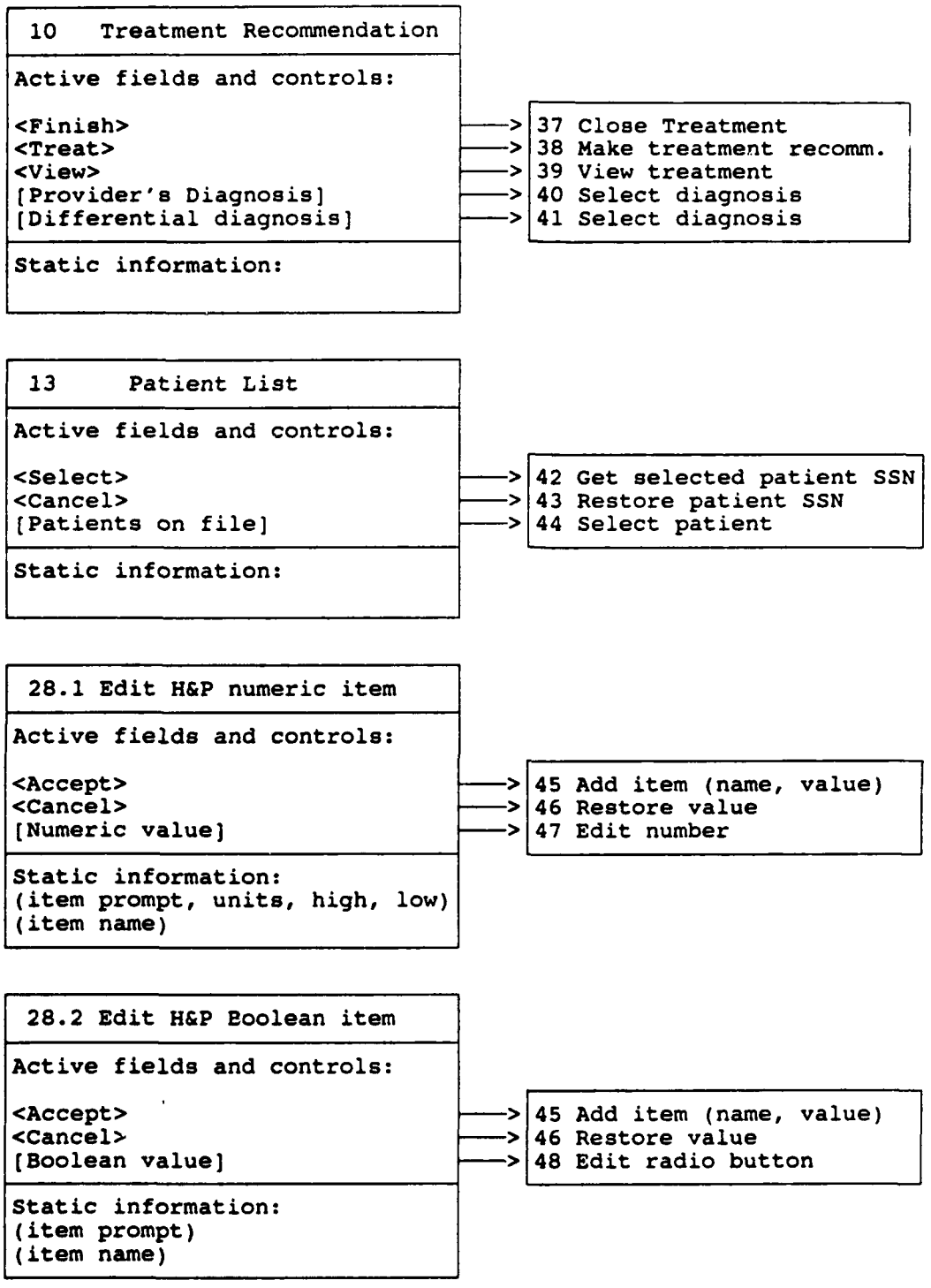


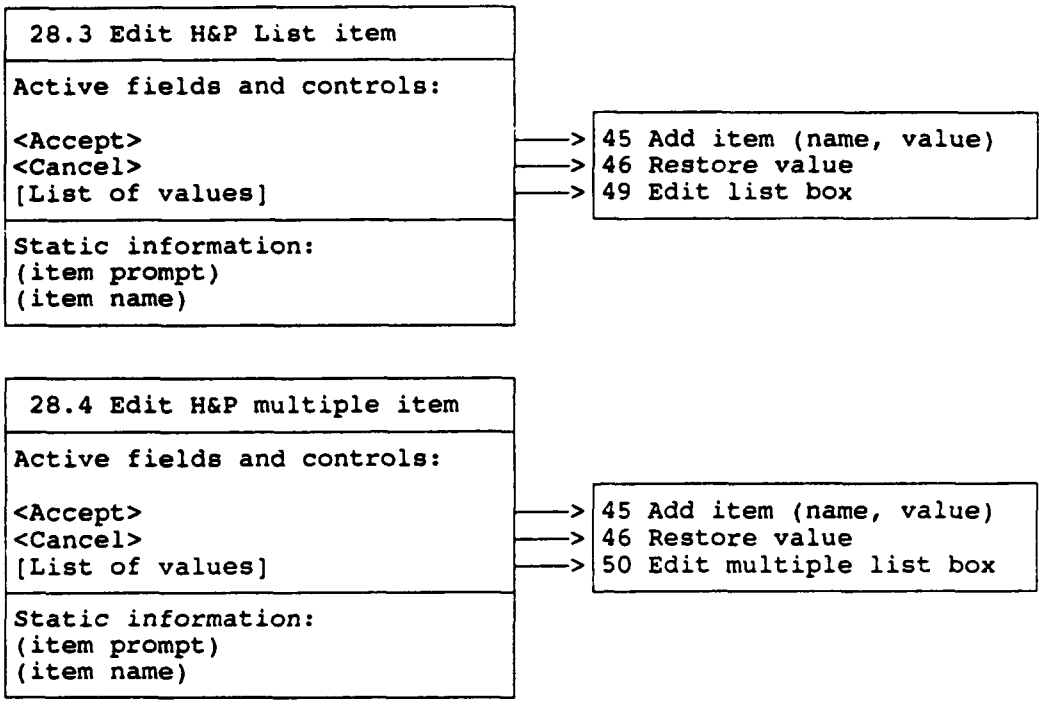
CAMD INTERACTIVE SCREEN OBJECTS




2 Encounter Screen
Active fields and controls:
<New>
<Close>
<Print>
[Complaint]
[Patient]
[History and Physical]
[Diagnosis]
[Assistance]
[Treatment]
Static information:
(Encounter date, time, status)
(Care provider)
3 Open encounter
4 Close encounter
5 Print encounter
6 Enter complaint
1 Register patient
7 History and Physical
8 Provider's diagnosis
9 Diagnostic assistance
10 Treatment recommendation
3 Open Encounter
Active fields and controls:
<New>
<Resume>
<Cancel>
[Pending encounters]
Static information:
17 Open new encounter
18 Resume selected encounter
19 Restore encounter
20 Select pending encounter
4 Close Encounter
Active fields and controls:
<Save>
<Abort>
<Resume>
Static information:
(Encounter date, time, status)
21 Save encounter
22 Abort encounter
23 Resume encounter
5 Print Encounter
Active fields and controls:
<Print>
<Cancel>
[Print options]
Static information:
(Encounter date, time, status)
51 Send encounter
52 Abort print
53 Set print options


6 Complaint
Active fields and controls:
<Select>
<Cancel>
[Complaints list]
Static information:
24 Enter selected complaint
25 Restore complaint
26 Select complaint
7 History and Physical
Active fields and controls:
<Finish>
<Resume>
<Edit>
[Hist/Phys list]
Static information:
27 Close H&P
28 Edit next H&P item
29 Edit selected H&P item
30 Select H&P item
8 Provider's Diagnosis
Active fields and controls:
<Select>
<View>
<Cancel>
[Diagnosis list]
Static information:
31 Enter selected diagnosis
32 View disease description
33 Restore diagnosis
34 Select diagnosis
9 Diagnostic Assistance
Active fields and controls:
<Finish>
<Calculate>
[Differential diagnosis]
Static information:
35 Close Diagnostic Assist
36 Make diagnostic recomm.


10 Treatment Recommendation
Active fields and controls:
<Finish>
<Treat>
<View>
[Provider's Diagnosis]
[Differential diagnosis]
Static information:
37 Close Treatment
38 Make treatment recomm.
39 View treatment
40 Select diagnosis
41 Select diagnosis
13 Patient List
Active fields and controls:
<Select>
<Cancel>
[Patients on file]
Static information:
42 Get selected patient SSN
43 Restore patient SSN
44 Select patient
28.1 Edit H&P numeric item
Active fields and controls:
<Accept>
<Cancel>
[Numeric value]
Static information:
(item prompt, units, high, low)
(item name)
45 Add item (name, value)
46 Restore value
47 Edit number
28.2 Edit H&P Boolean item
Active fields and controls:
<Accept>
<Cancel>
[Boolean value]
Static information:
(item prompt)
(item name)
45 Add item (name, value)
46 Restore value
48 Edit radio button


28.3 Edit H&P List item
Active fields and controls:
<Accept>
<Cancel>
[List of values]
Static information:
(item prompt)
(item name)
45 Add item (name, value)
46 Restore value
49 Edit list box
28.4 Edit H&P multiple item
Active fields and controls:
<Accept>
<Cancel>
[List of values]
Static information:
(item prompt)
(item name)
45 Add item (name, value)
46 Restore value
50 Edit multiple list box

APPENDIX C. CAMD EXPERT SYSTEM KNOWLEDGE BASE DEVELOPER'S GUIDE

CONTENTS

1. INTRODUCTION

This document is the knowledge base developer's guide. It describes the knowledge base and procedures for maintaining the rulebase and associated data structures. The diagnostic algorithm and data flows are briefly reviewed before describing the maintenance requirements. The initial version of the rulebase supports only the Bayesian diagnostic algorithm. Later versions will be able to support more general rule-based methods.

2. KNOWLEDGE BASE DEFINITION

To aid in understanding the maintenance process, it is necessary to look at the underlying databases and tables that support the knowledge base. The following tables are required by the Bayesian Diagnostic Engine. These tables are briefly described below.

2.1 AREA

This table lists the possible complaints or diagnostic areas that the engine can handle. An area entry consists of the following fields:

area	area identifier
name	area external name
threshold	also consider display threshold
probable	probable diagnosis display threshold
likely	likely diagnosis display threshold

The area field is the identifier for the diagnostic area. It is used internally to select a set of diagnoses for consideration by the diagnostic algorithm.

The name field is the (user oriented) name for the diagnostic area or complaint.

The threshold, probable, and likely fields are display thresholds for the diagnoses, allowing three levels of confidence in the computer's diagnoses to be presented to the user.

2.2 RULE

This table lists all of the diagnostic rules (Bayesian in this version). A rule consists of the following fields:

rule	rule identifier
area	rule segment selector
salience	rule salience
text	rule source code/rule documentation
premise	rule premise clause identifier
action	rule action clause identifier

The area field is used to select (filter) different rule sets out of the total set of rules. For example, abdominal pain would be one area and chest pain another, but in general, one could have many other segment selectors for more sophisticated diagnostic algorithms. For the Bayesian method, it is expected that just one rule set would be used per complaint area.

The rulebase is indexed on the salience field. This allows the developer to specify the order in which rules will be fired. The lower the salience, the higher the rule priority.

The rulebase is also indexed on the rule field. This allows rules to be selected for viewing or editing, or a particular rule to be fired.

The source code for the rule appears, with comments, in the text field.

The premise field is an identifier for premise clauses associated with the rule.

Similarly, the action field is an identifier for action clauses associated with the rule. Each rule can have any number of premise or action clauses associated with it. All clauses will have the same identifier. Note that this cannot be the same as the rule number, since the rule number can be changed at any time. Also, several rules can have the same clauses associated with them.

2.3 PREMISE

This table is the set of all clauses to be evaluated by the rules. Each rule can select several premises. All premises associated with the rule must evaluate to true in order for the rule to become eligible (placed on the agenda). There is, therefore, an implied AND operator joining the clauses. A premise consists of the following fields:

clause	premise identifier
op	operation code
object	object type descriptor
id	object identifier
val	value

The clause field is used to select the set of clauses associated with the rule.

The op field is the operation code (relational operator) that is to be applied.

The object field describes the type of object pertaining to the clause. Currently this can be either an "S" for sign/symptom

or "D" for disease. This object descriptor allows the appropriate sign/symptom or disease table to be selected for evaluating the clause.

The id field is the identifier of the particular object referenced by the clause. This id is used to retrieve the item from the dictionary or disease table.

The val field is the value against which the item will be evaluated.

NOTE: A premise is a simple expression with the following syntax:

<objectid> <op> <val>

where:

<objectid> is the name of a symptom or disease (as described in 2.5 DICT and 2.8 DISEASE below)

<op> is the operator which can be one of the following:

```
<
>
==
!=
<=
>=
=
```

<val> can be any number or string constant (string constants are defined in 2.6 ENUM below).

Examples:

```
TEMPERATURE > 102
COLOR == "PALE"
```

This is a very simple syntax, but it is sufficient to handle the type of expressions required in the Bayesian rulebase. Note that because of the implied AND, the syntax will support expressions of the form:

```
TEMPERATURE > 102 AND COLOR == "PALE"
```

2.4 ACTION

This action table is very similar to the premise table. The action clauses are evoked when the rule is selected from the agenda. Each action clause results in some operation being applied to the working memory, whereas premise clauses usually test the values in working memory. Apart from this difference,

the structure of the two clause types is almost identical. An action consists of the following fields:

clause	action clause identifier
op	operation code
object	object type descriptor
id	object identifier
val	value

The clause field is used to select the set of action clauses associated with the rule.

The op field is the operation code that is to be applied to the object.

The object field describes the type of object pertaining to the clause. Currently this can be either an "S" for sign/symptom or "D" for disease. This object descriptor allows the appropriate sign/symptom or disease table to be selected for evaluating the clause.

The id field is the identifier of the particular object referenced by the clause. This id is used to retrieve the item from the dictionary or disease table.

The val field is the value against which the item will be evaluated.

NOTE: An action clause, similar to a premise, is a simple expression with the following syntax:

<objectid> <op> <val>

Examples:

COLOR = "PALE"
APPENDICITIS = 58.3

2.5 DICT

This table describes each data element referenced by the diagnostic algorithm. A data element dictionary item consists of the following fields:

id	data element identifier
name	data element name
datatype	data element type
askable	user askable value
question	user question to solicit value

The id field is used by the premise/action clauses to select the item from the dictionary.

The name field is the (user oriented) short name for the data element. Examples are "TEMPERATURE" and "DIASTOLIC BP."

The datatype field is used to interpret the data element value. Possible data types are the following: Logical, numeric, enumerated, and multiple value.

The askable field determines whether the user can be asked to enter a value for the item. Nonaskable items, known as metaqualifiers, are generally computed values that may not have any meaning to the user. The system itself must derive the value from other known quantities.

The question field is used to ask the user for the value of the data element if it is missing. The question is triggered automatically by the diagnostic algorithm when it tries to evaluate a clause.

2.6 ENUM

This table lists the enumerated values for data elements of type enumerated or multiple value. An enumerated type consists of the following fields:

id	data element identifier
mutex	mutual exclusion attribute
ord	subelement identifier for the enum instance
enumerate	element value instance

The id field corresponds to the dictionary id referenced by this item. Each dictionary item can have several enum choices associated with it.

The mutex field is used for multiple selection types to determine mutual exclusion between certain user selections.

The enumerate field is the actual value of the element.

The ord field is a subelement identifier or ordinal for the enumerated value.

2.7 VAL

This table describes the attributes for numeric types.

The id field corresponds to the dictionary id referenced by this item. Each dictionary item can have only one val associated with it. A value type consists of the following fields:

id	data element identifier
width	element display/storage field width
dec	number of decimal places precision

lo	lower limit of range
hi	higher limit of range
units	engineering units

The width field describes the total width allocated to this data element. This limits both storage as well as display of this item.

The dec field is the number of decimal places of precision allowed for this item.

The lo and hi fields, respectively, give the lower and upper range of this data element. This range is used to validate the element when entered by the user or imported from an outside source. Either lo or hi left blank implies that there is no range check applied.

The units field describes the (user oriented) engineering units for the physical quantity represented by the item.

2.8 DISEASE

This table describes all diseases that the system can handle. It is used to retrieve disease descriptions, treatment descriptions, and to construct differential diagnoses. The diagnostic engine will select a subset of the diseases in this table in generating a diagnosis. A disease entry consists of the following fields:

id	disease identifier
name	disease name
descript	disease description

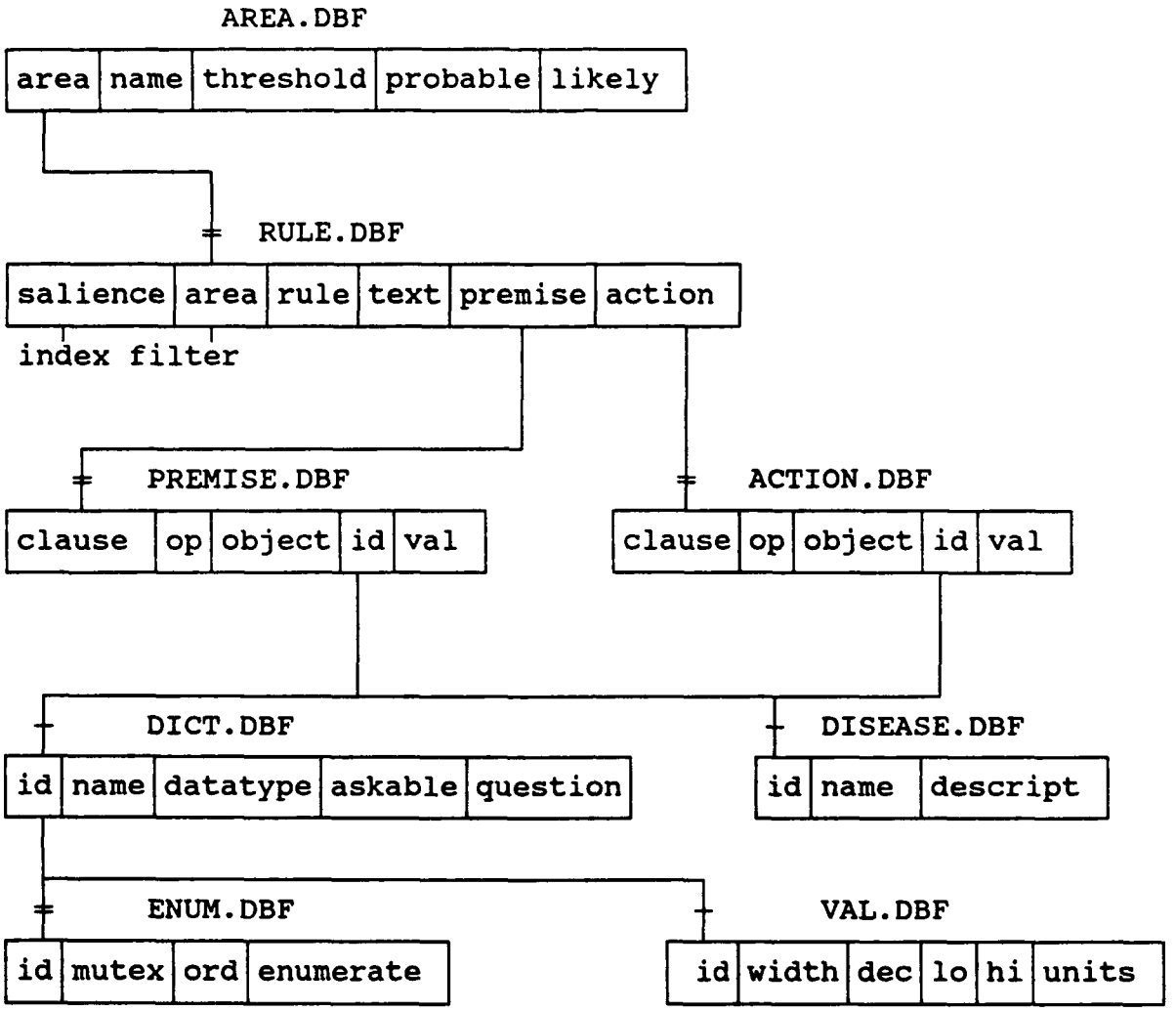
The id field is a unique identifier for the disease. It is used by the diagnostic engine to select diseases for the differential diagnosis.

The name field is the (user oriented) external name for the disease. It is used to display or print results only.

The descript field is the (user oriented) disease description. It is used to display, for the user, information about the disease and its treatment.

3. KNOWLEDGE BASE RELATIONSHIPS

The following charts show how the main rulebase tables are related to support the Bayesian Diagnostic Algorithm.



One to one relationships: single bar ┼.

One to many relationships: double bar ╪.

The chart shows the following important facts about the rulebase:

a. The AREA file is the primary parent of all other files (as far as the diagnostic engine is concerned).

b. The RULE file is indexed on the salience field. The salience field, therefore, is used to establish rule sequencing.

c. The RULE file is filtered on the area field. The area field, therefore, is used to establish a rule subset. The area is derived from the parent AREA file.

d. The PREMISE and ACTION files are child files of the rule file.

e. By setting up the the relations shown, the engine can select a particular rule and then retrieve corresponding PREMISE and ACTION clauses from the child files.

f. Since the RULE:PREMISE and RULE:ACTION relations are one-to-many, in general there will be several PREMISE and ACTION clauses associated with one rule.

g. The DICT and DISEASE files are child files of the PREMISE and ACTION files.

h. When a PREMISE or ACTION clause is selected, the corresponding DICT or DISEASE entry is selected. This is a one-to-one relationship.

i. Finally, the ENUM and VAL files are child files of the DICT file.

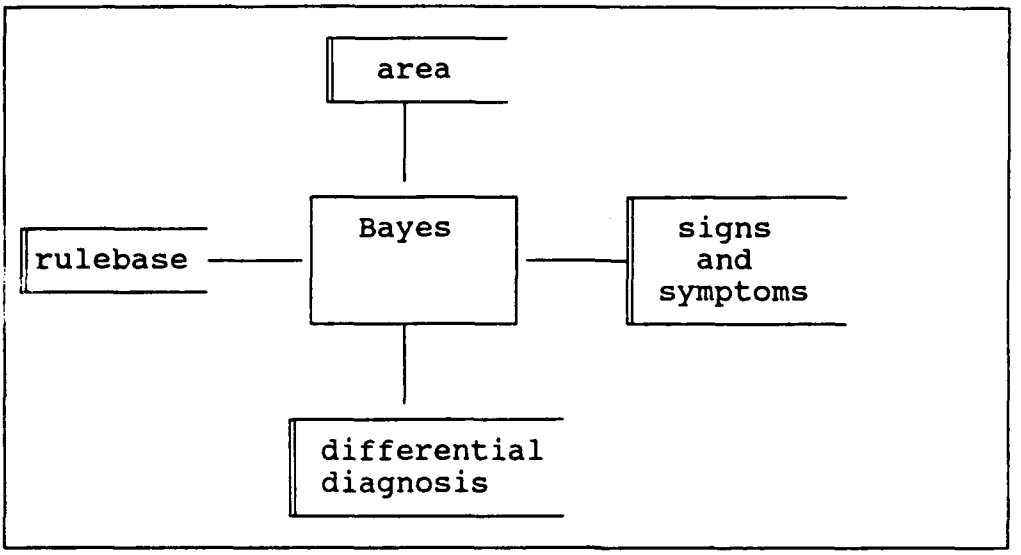
j. Each DICT element can have several ENUM entries associated with it, since this is a one-to-many relationship.

k. Each DICT element can have one VAL entry associated with it, since this is a one-to-one relationship.

4. DIAGNOSTIC ALGORITHM DATA FLOW

In this section, the data flows associated with the rulebase tables are described. It is necessary to review this process, so that the purpose of the database tables is clearly understood by the persons responsible for their maintenance.

4.1 Data Flow Diagram



The Data Flow Diagram shows the top level of the Bayes Algorithm. Note the following important facts from this diagram.

The diagram consists of several objects:

process	denoted by a closed rectangle
data store	denoted by an open rectangle, with double line side.
data flow	denoted by a line between objects. A flow originates at the box where it is attached terminates at the other box.

Processes:

"Bayes"	represents the Diagnostic algorithm. All of the Bayes Algorithm logic is encompassed in this process.

Data Stores:

The Bayes process reads from several data stores or databases and writes to several data stores. Note that the "signs and symptoms" and the "differential diagnosis" data stores together are referred to as Working Memory:

"area"	the diagnostic area that the algorithm is to address (e.g., abdominal, chest, occular)

"rulebase"	the rulebase tables driving the algorithm. The set of tables described in Section 2.
"signs and symptoms"	the physical signs and symptoms presented by the subject encounter.
"differ diagnosis"	the differential diagnosis computed by the algorithm.

Data Flows:

From:	To:
"Signs and Symptoms"	"Bayes"
"rulebase"	"Bayes"
"Bayes"	"Differential diagnosis"
"area"	"Bayes"

4.2 Bayes Algorithm

The algorithm is shown here using simple XBASE code. Most of this code is directly executable in FoxPro, but some of the programming details have been suppressed to make the logic more readable. The purpose of the code presented here is to illustrate the use of the database tables and the relationships required between tables.

```
PARAMETERS marea

SELECT 0
USE dict INDEX dictid
SELECT 0
USE premise INDEX premise
SET RELATION TO id INTO dict
SELECT 0
USE action INDEX action
SELECT 0
USE rule INDEX salience
SET FILTER TO area = marea
SET RELATION TO premise INTO premise, action INTO action

SCAN WHILE !eof()
   m.result = .t.
   SELECT premise

   SCAN WHILE clause = rule->premise
      IF !evalcond( object, id, op, val)
         m.result = .f.
         EXIT
      ENDIF
   ENDSCAN

   IF mresult
      SELECT action
      SCAN WHILE clause = rule->action
         evalcond( object, id,  op, val)
      ENDSCAN
   ENDIF
   SELECT rule
ENDSCAN

RETURN
```

NOTES:

evalcond() evaluates the premise/action clause.

This function computes expressions in the rulebase with formal parameters instantiated from the actual values found in the working memory. Note that this function operates directly on elements in working memory. If an undefined element is referenced and it is askable, a data entry window is triggered to get the value from the user. Recall that working memory is the aggregate of the signs and symptoms and diagnoses computed by the diagnostic engine.

5. KNOWLEDGE BASE MAINTENANCE

With the foregoing descriptions out of the way, we are now ready to describe the procedures for maintaining the knowledge base.

5.1 Knowledge Objects

As described in Section 5.2 below, the maintenance process is modeled in terms of knowledge objects and edit transactions on those objects. Knowledge objects consist of knowledge bases, rules, qualifiers, diseases, and so on. Each object is described in terms of a set of attributes. The attributes of an object are gathered in a structure or database. Most of these objects and their attributes have already been discussed in Sections 2 and 3, which describe the physical mapping or implementation of the knowledge base on the XBASE database layer. In this section we will describe the objects in more abstract terms:

5.1.1 Knowledge Base Object

A knowledge base consists of an entry in the AREA table and a set of associated rule objects.

5.1.2 Rule Object

A rule object consists of an entry in the RULE table and related entries in the PREMISE and ACTION tables. Attributes of a rule include a salience which determines the rule order. The PREMISE and ACTION attributes can refer to either qualifier or disease objects.

5.1.3 Disease Object

A disease object consists of an entry in the DISEASE table.

5.1.4 Qualifier Object

A qualifier object consists of an entry in the DICT table and related entries in the ENUM or VAL tables.

5.2 Knowledge Base Editor

Knowledge base maintenance consists of three primary edit transactions. These are Create, Edit, and Delete. Note that these transactions must maintain relational integrity (something that FoxPro and XBASE do not do). In other words, any operation on an object is not limited purely to the attributes of that object alone, but has to ensure that other related objects are properly maintained and remain valid after the edit transaction. Examples of these relational constraints will be covered in the appropriate descriptions below.

Note further that in order to maintain integrity of existing (user) transactions, some edit/delete transactions are not allowed at all. For example, if a past patient encounter has referenced a dictionary item, it can no longer be deleted nor can its identifier be altered. Certain attributes, for example the text used to ask the user for a value, can safely be edited.

5.2.1 Create Object

This transaction consists of defining or adding a new instance of a particular object to the knowledge base. In order to simplify the maintenance procedures and to ensure integrity is established immediately, objects have to be created in the following order.

a. create knowledge base
b. create disease
c. create qualifier
d. create rule

In general, the attributes of an object have to be defined (created) before they are used to define the object (this is analogous to defining variables in a program before they are used, or defining elements of a structure before defining the structure itself).

5.2.1.1 Create Knowledge Base

This step consists of defining an entry in the AREA database. Creating a knowledge base corresponds to adding a new diagnostic capability to the system (e.g., abdominal pain, chest pain, occular problems). Each diagnostic area has a set of diseases, qualifiers (symptoms), and rules associated with it. There can also be sets of metarules that, for example, decide to which diagnostic area the patient's complaint corresponds.

The knowledge base definition consists of the name and any other parameters that are unique to this particular area (e.g., thresholds). At this point, we have an empty knowledge base. Since no rules have been defined yet, nothing will happen when we run the system and select this knowledge base.

5.2.1.2 Create Disease

This step consists of defining any diseases that will be required to support the knowledge base. Since all diseases known to the system will appear in this list, some of the diseases will already be there. For example, if MYOCARDIAL INFARCTION was a diagnosis considered in the ABDOMINAL PAIN algorithm, and we are now defining a CHEST PAIN knowledge base that will also consider Myocardial Infarction, then this disease need not be defined again. The create disease procedure must define entries in the DISEASE database. The following attributes must be defined at a minimum. Other attributes need not be defined unless required by other applications:

id: disease identifier. Automatically assigned by the maintenance utility. Must be a unique number so that diseases can be referenced by id rather than by name.

disease: disease name. Can be entered as text by the knowledge base maintainer.

5.2.1.3 Create Qualifier

This step consists of defining any qualifiers, or data elements that will be required to support the knowledge base. As in the case of diseases above, data elements may in general already appear in the data element dictionary if they are referenced by another knowledge base. Only elements not yet defined must now be created. The create qualifier procedure must define entries in the DICT database and, depending on the data type, entries in the ENUM or VAL databases. The following attributes must be defined:

id: data element identifier. Automatically defined by the system so that data elements can be referenced by id rather than by name.

name: data element name. Short meaningful name entered by the user. This name will be used to display data elements in reports presented to the user. The user can use this name to retrieve information about the element.

datatype: data element type. Must be chosen from a strictly defined list of types allowed by the system. User can select one item from the list. The following choices are allowed:

L logical
N numeric
E enumerated
M multiple choice

askable: set to true if user can be asked for a value, false if user cannot be asked for a value.

question: user question to solicit value. User can enter any text here.

For enumerated types (ENUM database):

id: automatically set to the same as dict id by the maintenance utility.

mutex: mutual exclusion attribute. Automatically set to blank by system (not required in this table).

enumerate: element value instance. The user can enter any short text phrase here.

ord: element value ordinal. A number automatically set by the system identifying the text name entered for the enumerate field.

For multiple choice types (ENUM database):

id: automatically set same as dict id by the system.

mutex: mutual exclusion attribute. Set to one of the following:

- if this choice excludes all other choices.
+ if this choice does not exclude other choices.

enumerate: element value instance. The user can enter any short text phrase here.

ord: element value ordinal. A number automatically set by the system identifying the text name entered for the enumerate field.

For numeric types (VAL database):

width: element display/storage field width.

dec: number of decimal places precision.

lo: lower limit of range. Leave blank if no limit required.

hi: higher limit of range. Leave blank if no limit required.

units: engineering units. Any short text phrase can be entered here.

5.2.1.4 Create Rule

Before any rules can be created, the above three procedures must have been completed. A rule must be associated with a known knowledge base. Moreover, any diseases or qualifiers that it references must already have been defined. The following attributes of a rule must be defined:

- rule: user can enter any number here to identify the rule.
- salience: user can enter any number here to define the rule order.
- area: must be selected from the area table.
- text: rule source code, filled in automatically by the system from the set of clauses, as described below. User can enter any comments about the rule.
- premise: filled in automatically by the system.
- action: filled in automatically by the system.

At least one premise and one action clause must be defined for each rule. For clauses, the following attributes must be defined:

- clause: automatically copied by the system from the premise or action field of the rule.

Each clause is entered by the user using the syntax described in Sections 2.3 and 2.4 above:

<object> <op> <value>

The system will translate this syntax, automatically look up the names in the dictionary, and fill in the following fields accordingly:

op:
object:
id:
val:

5.2.2 Edit Object

This transaction consists of editing the attributes of an existing knowledge base object. As discussed above, certain constraints have to be applied to maintain database integrity.

5.2.2.1 Edit Knowledge Base

The following fields can be changed:

name:
threshold:
probable:
likely:

The following fields can not be changed:

area:

5.2.2.2 Edit Disease

The following fields can be changed:

descript:

The following fields can not be changed:

id:
name:

5.2.2.3 Edit Qualifier

The following fields can be changed:

question:
mutex:
width:
dec:
lo:
hi:
units:

The following fields can not be changed:

id:
name:
datatype:
ord:
enumerate: One can add additional enumerate items, but one can not delete items.

5.2.2.4 Edit Rule

All fields can be changed.

5.2.3 Delete Object

This transaction consists of erasing an existing object. As discussed above, certain objects cannot be deleted because this action does not maintain database integrity.

5.2.3.1 Delete Knowledge Base

A knowledge base can not be deleted. It can be emptied, but only in a restricted way. The entry in the AREA table can not be removed. However, all rules associated with the knowledge base can be deleted.

5.2.3.2 Delete Disease

A disease can not be deleted.

5.2.3.3 Delete Qualifier

A qualifier can not be deleted.

5.2.3.4 Delete Rule

A rule can be deleted. When this is done, all premise and action clauses associated with the rule are also deleted.

5.3 Knowledge Base Editor User Interface

The following table suggests a possible knowledge base editor user interface. It shows the editor screen, with a particular knowledge base selected and threshold parameters filled in. A certain rule has been selected, and its clauses are shown in the boxes at the lower left. At the lower right are shown boxes containing the data element primitives, operators, and goals. When the user edits a clause, elements can be picked from the primitives in the right hand boxes.

<table>
<tr><td colspan="4">Knowledge Base Editor</td></tr>
<tr><td colspan="4">Knowledge base name: [Abdominal Pain]</td></tr>
<tr><td colspan="2">Display thresholds: Consider[0]</td><td>Probable [10]</td><td>Likely [50]</td></tr>
<tr><td colspan="2">Rule [1] [Salience]</td><td colspan="2">Operators</td></tr>
<tr><td colspan="2">Premises</td><td colspan="2">== > < >= <= != = $</td></tr>
<tr><td>Object
[...]</td><td>Op
[.]</td><td>Value
[...]</td><td colspan="2" rowspan="2">Data Elements
Object [...] | Datatype [...]</td></tr>
<tr><td colspan="2">Actions</td></tr>
<tr><td>Object
[...]</td><td>Op
[.]</td><td>Value
[...]</td><td colspan="2">Goals
Object [...]</td></tr>
</table>

5.4 Knowledge Base Loader

The knowledge base loader is a noninteractive, batch-oriented utility that loads a complete knowledge base definition from an external file. The same operations described for the editor above are done by the loader, except that the operations are triggered by batch commands and knowledge object descriptions in a user-generated external file. The loader syntactically checks the knowledge base description and validates any operations before updating the knowledge databases.

The loader is divided into two phases as follows:

The first phase syntactically checks the user descriptions and translates the description file into a set of intermediate knowledge base transaction files. These files have the same structure as the final database files. Some fields are not instantiated until the phase two semantic check is complete.

The second phase semantically checks the proposed transactions for validity and updates the knowledge base from the intermediate file.

5.4.1 Knowledge Base Syntax

The following is a formal description of the knowledge base syntax accepted by the parser (phase one of the knowledge base loader):

Literals are shown as uppercase text.

Tokens are shown as lowercase text, and a token is defined elsewhere in the description. Special characters are taken as literals, except for the following:

[ ]	surround an optional part of a token descrition.
¦	separate alternative forms of a token.

rulebase:

RULEBASE name
THRESHOLD number PROBABLE number LIKELY number
goals qualifiers rules

qualifiers:

qualifier [qualifier]

```
qualifier:

     name
     [ NUMERIC numattributes
     ¦ LOGICAL ( enumlist )
     ¦ ENUM ( enumlist )
     ¦ MULTIPLE ( multlist )]
     [ prompts ] ;

goals:

     goal [ goal ]

goal:

     GOAL name
     [ DESCRIPTION memo ]
     [ TREATMENT memo ]
     [ BRIEF memo ]

numattributes:

        WIDTH  number
     ¦  DEC    number
     ¦  RANGE  number TO number
     ¦  UNITS  string

enumlist:

     enumitem [ , enumitem ]

multlist:

     mutex [ , mutex ]

mutex:

     mutexattribute enumitem

enumitem:

     string [ REPORT string ]

mutextattribute:

     [ + ¦ - ]

prompts:

     [ QUESTION string ]
     [ PROMPT string ]
     [ REPORT string ]
     [ HELP memo ]
```

```
rules:

        RULE rulenumber salience rulebody
      | comment
      | EOF

rulebody:

     rulestatement

rulestatement:

     IF condition THEN action END

action:

     statement [ ; statement ]

statement:

     object = constant

condition:

     logicalterm

logicalterm:

     logicalfactor [ AND logicalfactor ]

logicalfactor:

     expression

expression:

     object [ < | <= | == | != | >= | > constant ]

object:

     name

name:

     string

constant:

        number
      | string
```

memo:

string

number:

digit [digit]

string:

" char [char] "

comment:

anychar

5.4.2 Knowledge Base Validation

The validity checks performed are the following:

Duplicate object checks

Checks whether object to be appended to the data base already exists. If it exists, takes the transaction as an update rather than as a create operation. Only allowable fields are updated.

Completeness checks

Checks that all required attributes of an object have been defined.

Consistency checks

Checks that all attributes of an object are consistent with its definition and type.

5.4.3 Example Definition

The following definition is a complete example of the syntax described above.

```
RULEBASE "Acute abdominal pain"
   THRESHOLD  5 PROBABLE 45 LIKELY 75

GOAL "APPENDICITIS"
GOAL "NON SPECIFIC ABDOMINAL PAIN"
GOAL "RENAL COLIC"
GOAL "PERFORATED DUODENAL ULCER"
GOAL "CHOLECYSTITIS"
GOAL "SMALL BOWEL OBSTRUCTION"
```

```
GOAL "PEPTIC ULCER DISEASE"
GOAL "MESENTERIC ADENITIS"
GOAL "DIVERTICULITIS"
GOAL "INGUINAL HERNIA"
GOAL "ACUTE PANCREATITIS"
GOAL "MYOCARDIAL INFARCTION"
GOAL "RIGHT LOWER LOBE PNEUMONIA"
GOAL "PANCREATITIS"
GOAL "GASTRITIS"
GOAL "GASTROENTERITIS"
GOAL "HEPATITIS"

MULTIPLE  "SITE AT ONSET"
   (+ "RIGHT UPPER QUAD",
   + "LEFT UPPER QUAD",
   + "RIGHT LOWER QUAD",
   + "LEFT LOWER QUAD",
   + "UPPER HALF",
   + "LOWER HALF",
   + "RIGHT HALF",
   + "LEFT HALF",
   + "CENTRAL",
   + "GENERAL",
   + "RIGHT FLANK",
   + "LEFT FLANK",
   - "NO PAIN AT ONSET")
   QUESTION "Which of the following best describes the site of
the pain at the ONSET"

ENUM  "CHARACTER OF PAIN"
   ("INTERMITTENT",
    "STEADY",
    "COLICKY")
   QUESTION "How does your patient describe the CHARACTER OF
PAIN"

ENUM  "INTENSITY OF PAIN"
   ("MODERATE",
    "SEVERE")
   QUESTION "How do you rate the INTENSITY OF PAIN"

MULTIPLE  "AGGRAVATING FACTORS"
   (+ "MOVEMENT",
   + "COUGH",
   + "BREATHING",
   + "FOOD",
   + "OTHER",
   - "NONE")
   QUESTION "Do any of the following aggravate the pain"
```

```
ENUM  "PROGRESS OF PAIN"
   ("BETTER",
    "SAME",
    "WORSE")
   QUESTION "How does your patient rate the PROGRESS OF PAIN"

MULTIPLE  "RELIEVING FACTORS"
   (+ "LYING STILL",
   + "VOMITING",
   + "ANTACIDS",
   + "FOOD",
   + "OTHER",
   - "NONE")
   QUESTION "Do any of the following relieve the pain"

MULTIPLE  "BOWELS"
   (- "NORMAL",
   - "CONSTIPATED",
   - "DIARRHEA",
   - "BLOOD IN STOOLS",
   - "MUCUS IN STOOLS")
   QUESTION "The inspection of the BOWELS reveals"

MULTIPLE  "URINE"
   (- "NORMAL",
   + "FREQUENT",
   + "PAIN WHEN URINATING",
   + "DARK (BILE)",
   + "BLOOD IN URINE")
   QUESTION "The inspection of the URINE reveals"

ENUM  "ABDOMINAL MOVEMENT"
   ("NORMAL",
    "PERISTALSIS",
    "DECREASED")
   QUESTION "The inspection of the ABDOMEN reveals"

ENUM  "BOWEL SOUNDS"
   ("NORMAL",
    "ABSENT",
    "HYPERACTIVE")
   QUESTION "Listening indicates BOWEL SOUNDS are"

MULTIPLE  "TENDERNESS"
   (+ "RIGHT UPPER QUAD",
   + "LEFT UPPER QUAD",
   + "RIGHT LOWER QUAD",
   + "LEFT LOWER QUAD",
   + "UPPER HALF",
   + "LOWER HALF",
   + "RIGHT HALF",
   + "LEFT HALF",
```

```
   + "CENTRAL",
   + "GENERAL",
   + "RIGHT FLANK",
   + "LEFT FLANK",
   - "NONE")
   QUESTION "The patient indicates most TENDERNESS at the
following site"

ENUM  "RECTAL EXAM"
   ("NORMAL",
    "SHOWS MASS",
    "LEFT RECTAL TENDERNESS",
    "RIGHT RECTAL TENDERNESS",
    "GENERALIZED")
   QUESTION "The results of the RECTAL EXAM are"

MULTIPLE  "SITE AT PRESENT"
   (+ "RIGHT UPPER QUAD",
   + "LEFT UPPER QUAD",
   + "RIGHT LOWER QUAD",
   + "LEFT LOWER QUAD",
   + "UPPER HALF",
   + "LOWER HALF",
   + "RIGHT HALF",
   + "LEFT HALF",
   + "CENTRAL",
   + "GENERAL",
   + "RIGHT FLANK",
   + "LEFT FLANK",
   - "NO PAIN AT PRESENT")
   QUESTION "Which of the following describes the SITE of PAIN AT
PRESENT"

ENUM  "COLOR"
   ("NORMAL",
    "PALE",
    "FLUSHED",
    "JAUNDICED",
    "CYANOTIC")
   QUESTION "Which of the following best describes the COLOR of
the patient"

ENUM  "MOOD"
   ("NORMAL",
    "DISTRESSED",
    "ANXIOUS")
   QUESTION "Which of the following best describes the MOOD of
the patient"

ENUM  "SEX"
   ("MALE",
    "FEMALE")
   PROMPT "SEX"
```

```
LOGICAL  "DISTRESS"
   ("YES", "NO")
   QUESTION "Does the patient show signs of DISTRESS"

LOGICAL  "NAUSEA"
   ("YES", "NO")
   QUESTION "Does the patient complain of NAUSEA"

LOGICAL  "VOMITING"
   ("YES", "NO")
   QUESTION "Has the patient been VOMITING"

LOGICAL  "APPETITE"
   ("YES", "NO")
   QUESTION "Has the APPETITE of the patient decreased"

LOGICAL  "JAUNDICE"
   ("YES", "NO")
   QUESTION "Does the skin of the patient have appearance of
JAUNDICE"

LOGICAL  "INDIGESTION"
   ("YES", "NO")
   QUESTION "Does the patient complain of previous INDIGESTION"

LOGICAL  "SIMILAR PAIN"
   ("YES", "NO")
   QUESTION "Has the patient had previous SIMILAR PAIN"

LOGICAL  "ABDOMINAL SURGERY"
   ("YES", "NO")
   QUESTION "Has the patient had ABDOMINAL SURGERY"

LOGICAL  "PREVIOUS ILLNESS"
   ("YES", "NO")
   QUESTION "Does the patient have a history of pertinent
PREVIOUS ILLNESS"

LOGICAL  "MEDICATION"
   ("YES", "NO")
   QUESTION "Is the patient taking MEDICATION"

LOGICAL  "ABDOMINAL SCARS"
   ("YES", "NO")
   QUESTION "Are ABDOMINAL SCARS PRESENT"

LOGICAL  "GUARDING"
   ("YES", "NO")
   QUESTION "Does the patient exhibit GUARDING"

LOGICAL  "RIGIDITY"
   ("YES", "NO")
   QUESTION "Does the patient exhibit RIGIDITY"
```

```
LOGICAL  "DISTENSION"
   ("YES", "NO")
   QUESTION "Is the abdomen of the patient distended (generalized
swelling)"

LOGICAL  "SWELLING"
   ("YES", "NO")
   QUESTION "Is there a localized SWELLING or mass"

LOGICAL  "MURPHY'S SIGN"
   ("YES", "NO")
   QUESTION "Is MURPHY'S SIGN present"

LOGICAL  "REBOUND TENDERNESS"
   ("YES", "NO")
   QUESTION "Is REBOUND TENDERNESS PRESENT"

LOGICAL  "APRIORI"
   ("YES", "NO")

NUMERIC  "WHITE BLOOD COUNT"
   WIDTH 5
   DEC 0
   RANGE 5.0 TO 10.0
   UNITS "10^3"
   QUESTION "What is the WHITE BLOOD COUNT"

NUMERIC  "DURATION OF PAIN"
   WIDTH 2
   DEC 0
   RANGE 1 TO 72
   UNITS "Hours"
   QUESTION "How long has the patient had the PAIN"

NUMERIC  "TEMPERATURE"
   WIDTH 5
   DEC 1
   RANGE 95 TO 106
   UNITS "Degrees F"
   QUESTION  "What is the TEMPERATURE of the patient"

NUMERIC  "PULSE"
   WIDTH 2
   DEC 0
   RANGE 30 TO 200
   UNITS "Per minute"
   QUESTION  "What is the PULSE rate of the patient"
```

```
NUMERIC  "RESPIRATION"
   WIDTH 2
   DEC 0
   RANGE 1 TO 60
   UNITS ""
   QUESTION  "What is the RESPIRATION rate of the patient"

NUMERIC  "SYSTOLIC"
   WIDTH 3
   DEC 0
   RANGE 50 TO 200
   UNITS "mm Hg"
   QUESTION  "What is the SYSTOLIC blood pressure of the patient"

NUMERIC  "DIASTOLIC"
   WIDTH 3
   DEC 0
   RANGE 50 TO 200
   UNITS "mm Hg"
   QUESTION  "What is the DIASTOLIC blood pressure of the
patient"

NUMERIC  "AGE"
   WIDTH 2
   DEC 0
   RANGE 16 TO 55
   UNITS "Years"
   PROMPT   "AGE"

NUMERIC  "HEIGHT"
   WIDTH 2
   DEC 0
   RANGE 60 TO 80
   UNITS "Inches"
   QUESTION  "What is the HEIGHT of the patient"

NUMERIC  "WEIGHT"
   WIDTH 3
   DEC 0
   RANGE 90 TO 300
   UNITS "Pounds"
   QUESTION  "What is the WEIGHT of the patient"
```

```
RULE 1 [10]
IF
"SEX                  "  ==  "M"
THEN
"APPENDICITIS                  " =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                   " =    0.1 ;
"PERFORATED DUODENAL ULCER     " =    0.1 ;
"CHOLECYSTITIS                 " =    0.1 ;
"SMALL BOWEL OBSTRUCTION       " =    0.1 ;
"PEPTIC ULCER DISEASE          " =    0.1
END

RULE 2 [10]
IF
"SEX                  "  ==  "F"
THEN
"APPENDICITIS                  "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                   "  =    0.1 ;
"PERFORATED DUODENAL ULCER     "  =    0.1 ;
"CHOLECYSTITIS                 "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION       "  =    0.1 ;
"PEPTIC ULCER DISEASE          "  =    0.1
END

RULE 3 [10]
IF
"AGE                  "  >= 0  AND
"AGE                  "  <= 9
THEN
"APPENDICITIS                  "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                   "  =    0.1 ;
"PERFORATED DUODENAL ULCER     "  =    0.1 ;
"CHOLECYSTITIS                 "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION       "  =    0.1 ;
"PEPTIC ULCER DISEASE          "  =    0.1
END

RULE 4 [10]
IF
"AGE                  "  >= 10   AND
"AGE                  "  <= 19
THEN
"APPENDICITIS                  "  =   25.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   19.0 ;
"RENAL COLIC                   "  =    5.0 ;
"PERFORATED DUODENAL ULCER     "  =    8.0 ;
"CHOLECYSTITIS                 "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION       "  =    8.0 ;
"PEPTIC ULCER DISEASE          "  =   12.0
END
```

```
RULE 5 [10]
IF
"AGE                    "  >= 20    AND
"AGE                    "  <= 29
THEN
"APPENDICITIS                    "  =    48.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    51.0 ;
"RENAL COLIC                     "  =    19.0 ;
"PERFORATED DUODENAL ULCER  "  =    16.0 ;
"CHOLECYSTITIS                   "  =     8.0 ;
"SMALL BOWEL OBSTRUCTION    "  =    16.0 ;
"PEPTIC ULCER DISEASE            "  =    38.0
END

RULE 6 [10]
IF
"AGE                    "  >= 30    AND
"AGE                    "  <= 39
THEN
"APPENDICITIS                    "  =    15.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =     9.0 ;
"RENAL COLIC                     "  =    32.0 ;
"PERFORATED DUODENAL ULCER  "  =    14.0 ;
"CHOLECYSTITIS                   "  =    23.0 ;
"SMALL BOWEL OBSTRUCTION    "  =    16.0 ;
"PEPTIC ULCER DISEASE            "  =    21.0
END

RULE 7 [10]
IF
"AGE                    "  >= 40    AND
"AGE                    "  <= 49
THEN
"APPENDICITIS                    "  =     7.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    17.0 ;
"RENAL COLIC                     "  =    33.0 ;
"PERFORATED DUODENAL ULCER  "  =    32.0 ;
"CHOLECYSTITIS                   "  =    35.0 ;
"SMALL BOWEL OBSTRUCTION    "  =    20.0 ;
"PEPTIC ULCER DISEASE            "  =    23.0
END
```

```
RULE 8 [10]
IF
"AGE                  "  >= 50   AND
"AGE                  "  <= 59
THEN
"APPENDICITIS                 "  =    6.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    4.0 ;
"RENAL COLIC                  "  =   11.0 ;
"PERFORATED DUODENAL ULCER    "  =   30.0 ;
"CHOLECYSTITIS                "  =   34.0 ;
"SMALL BOWEL OBSTRUCTION      "  =   40.0 ;
"PEPTIC ULCER DISEASE         "  =    6.0
END

RULE 9 [10]
IF
"AGE                  "  >= 60   AND
"AGE                  "  <= 69
THEN
"APPENDICITIS                 "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                  "  =    0.1 ;
"PERFORATED DUODENAL ULCER    "  =    0.1 ;
"CHOLECYSTITIS                "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION      "  =    0.1 ;
"PEPTIC ULCER DISEASE         "  =    0.1
END

RULE 10 [10]
IF
"AGE                  "  >  69
THEN
"APPENDICITIS                 "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                  "  =    0.1 ;
"PERFORATED DUODENAL ULCER    "  =    0.1 ;
"CHOLECYSTITIS                "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION      "  =    0.1 ;
"PEPTIC ULCER DISEASE         "  =    0.1
END

RULE 11 [90]
IF
"SITE AT ONSET        "  ==  "RIGHT UPPER QUAD"
THEN
"APPENDICITIS                 "  =    3.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    1.0 ;
"RENAL COLIC                  "  =    0.1 ;
"PERFORATED DUODENAL ULCER    "  =    6.0 ;
"CHOLECYSTITIS                "  =   38.0 ;
"SMALL BOWEL OBSTRUCTION      "  =    2.0 ;
"PEPTIC ULCER DISEASE         "  =   12.0
END
```

```
RULE 12 [90]
IF
"SITE AT ONSET      "  ==  "UPPER QUAD"
THEN
"APPENDICITIS                "  =    1.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    3.0 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    2.0 ;
"SMALL BOWEL OBSTRUCTION     "  =    2.0 ;
"PEPTIC ULCER DISEASE        "  =    8.0
END

RULE 13 [90]
IF
"SITE AT ONSET      "  ==  "RIGHT LOWER QUAD"
THEN
"APPENDICITIS                "  =   19.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   14.0 ;
"RENAL COLIC                 "  =   14.0 ;
"PERFORATED DUODENAL ULCER   "  =    3.0 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    2.0 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 14 [90]
IF
"SITE AT ONSET      "  ==  "LEFT LOWER QUAD"
THEN
"APPENDICITIS                "  =    2.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    9.0 ;
"RENAL COLIC                 "  =   11.0 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    2.0 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 15 [90]
IF
"SITE AT ONSET      "  ==  "UPPER HALF"
THEN
"APPENDICITIS                "  =   10.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   20.0 ;
"RENAL COLIC                 "  =    1.0 ;
"PERFORATED DUODENAL ULCER   "  =   59.0 ;
"CHOLECYSTITIS               "  =   45.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   28.0 ;
"PEPTIC ULCER DISEASE        "  =   58.0
END
```

```
RULE 16 [90]
IF
"SITE AT ONSET      "  ==  "LOWER HALF"
THEN
"APPENDICITIS                "  =    5.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   12.0 ;
"RENAL COLIC                 "  =    7.0 ;
"PERFORATED DUODENAL ULCER   "  =    4.0 ;
"CHOLECYSTITIS               "  =    2.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   20.0 ;
"PEPTIC ULCER DISEASE        "  =    6.0
END

RULE 17 [90]
IF
"SITE AT ONSET      "  ==  "RIGHT HALF"
THEN
"APPENDICITIS                "  =    2.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    6.0 ;
"RENAL COLIC                 "  =   18.0 ;
"PERFORATED DUODENAL ULCER   "  =    3.0 ;
"CHOLECYSTITIS               "  =    3.0 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    3.0
END

RULE 18 [90]
IF
"SITE AT ONSET      "  ==  "LEFT HALF"
THEN
"APPENDICITIS                "  =    1.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    4.0 ;
"RENAL COLIC                 "  =    8.0 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 19 [90]
IF
"SITE AT ONSET      "  ==  "CENTRAL"
THEN
"APPENDICITIS                "  =   49.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   29.0 ;
"RENAL COLIC                 "  =    1.0 ;
"PERFORATED DUODENAL ULCER   "  =   12.0 ;
"CHOLECYSTITIS               "  =   11.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   46.0 ;
"PEPTIC ULCER DISEASE        "  =   12.0
END
```

```
RULE 20 [90]
IF
"SITE AT ONSET       "  ==  "GENERAL"
THEN
"APPENDICITIS                "  =   10.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    4.0 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =   14.0 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    6.0 ;
"PEPTIC ULCER DISEASE        "  =    2.0
END

RULE 21 [90]
IF
"SITE AT ONSET       "  ==  "RIGHT FLANK"
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                 "  =   18.0 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 22 [90]
IF
"SITE AT ONSET       "  ==  "LEFT FLANK"
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    1.0 ;
"RENAL COLIC                 "  =   26.0 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 23 [90]
IF
"SITE AT ONSET       "  ==  "NO PAIN AT ONSET"
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                 "  =    1.0 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END
```

```
RULE 24 [100]
IF
"SITE AT PRESENT     "  ==  "RIGHT UPPER QUAD"
THEN
"APPENDICITIS                "  =    1.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    3.0 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    2.0 ;
"CHOLECYSTITIS               "  =   42.0 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =   12.0
END

RULE 25 [100]
IF
"SITE AT PRESENT     "  ==  "LEFT UPPER QUAD"
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    2.0 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    1.0 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    6.0
END

RULE 26 [100]
IF
"SITE AT PRESENT     "  ==  "RIGHT LOWER QUAD"
THEN
"APPENDICITIS                "  =   68.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   25.0 ;
"RENAL COLIC                 "  =   14.0 ;
"PERFORATED DUODENAL ULCER   "  =    2.0 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    2.0 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 27 [100]
IF
"SITE AT PRESENT     "  ==  "LEFT LOWER QUAD"
THEN
"APPENDICITIS                "  =    1.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    5.0 ;
"RENAL COLIC                 "  =   15.0 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    8.0 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END
```

```
RULE 28 [100]
IF
"SITE AT PRESENT     "  ==  "UPPER HALF"
THEN
"APPENDICITIS                 "  =    2.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   17.0 ;
"RENAL COLIC                  "  =    3.0 ;
"PERFORATED DUODENAL ULCER    "  =   46.0 ;
"CHOLECYSTITIS                "  =   42.0 ;
"SMALL BOWEL OBSTRUCTION      "  =   22.0 ;
"PEPTIC ULCER DISEASE         "  =   56.0
END

RULE 29 [100]
IF
"SITE AT PRESENT     "  ==  "LOWER HALF"
THEN
"APPENDICITIS                 "  =    7.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   12.0 ;
"RENAL COLIC                  "  =    5.0 ;
"PERFORATED DUODENAL ULCER    "  =    1.0 ;
"CHOLECYSTITIS                "  =    2.0 ;
"SMALL BOWEL OBSTRUCTION      "  =   14.0 ;
"PEPTIC ULCER DISEASE         "  =    2.0
END

RULE 30 [100]
IF
"SITE AT PRESENT     "  ==  "RIGHT HALF"
THEN
"APPENDICITIS                 "  =    4.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    3.0 ;
"RENAL COLIC                  "  =   14.0 ;
"PERFORATED DUODENAL ULCER    "  =   11.0 ;
"CHOLECYSTITIS                "  =    2.0 ;
"SMALL BOWEL OBSTRUCTION      "  =    0.1 ;
"PEPTIC ULCER DISEASE         "  =    2.0
END

RULE 31 [100]
IF
"SITE AT PRESENT     "  ==  "LEFT HALF"
THEN
"APPENDICITIS                 "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    2.0 ;
"RENAL COLIC                  "  =    9.0 ;
"PERFORATED DUODENAL ULCER    "  =    0.1 ;
"CHOLECYSTITIS                "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION      "  =    0.1 ;
"PEPTIC ULCER DISEASE         "  =    0.1
END
```

```
RULE 32 [100]
IF
"SITE AT PRESENT     "  ==  "CENTRAL"
THEN
"APPENDICITIS                "  =   14.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   21.0 ;
"RENAL COLIC                 "  =    1.0 ;
"PERFORATED DUODENAL ULCER   "  =    7.0 ;
"CHOLECYSTITIS               "  =    9.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   40.0 ;
"PEPTIC ULCER DISEASE        "  =   14.0
END

RULE 33 [100]
IF
"SITE AT PRESENT     "  ==  "GENERAL"
THEN
"APPENDICITIS                "  =    3.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    3.0 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =   34.0 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =   14.0 ;
"PEPTIC ULCER DISEASE        "  =    3.0
END

RULE 34 [100]
IF
"SITE AT PRESENT     "  ==  "RIGHT FLANK"
THEN
"APPENDICITIS                "  =    1.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    1.0 ;
"RENAL COLIC                 "  =   20.0 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 35 [100]
IF
"SITE AT PRESENT     "  ==  "LEFT FLANK"
THEN
"APPENDICITIS                "  =    1.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    1.0 ;
"RENAL COLIC                 "  =   26.0 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END
```

```
RULE 36 [100]
IF
"SITE AT PRESENT    "  ==  "NO PAIN AT PRESENT"
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    7.0 ;
"RENAL COLIC                 "  =    5.0 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    3.0 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    6.0
END

RULE 37 [110]
IF
"CHARACTER OF PAIN  "  ==  "INTERMITTENT"
THEN
"APPENDICITIS                "  =    5.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   18.0 ;
"RENAL COLIC                 "  =   14.0 ;
"PERFORATED DUODENAL ULCER   "  =    5.0 ;
"CHOLECYSTITIS               "  =    2.0 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =   14.0
END

RULE 38 [110]
IF
"CHARACTER OF PAIN  "  ==  "STEADY"
THEN
"APPENDICITIS                "  =   80.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   46.0 ;
"RENAL COLIC                 "  =   34.0 ;
"PERFORATED DUODENAL ULCER   "  =   86.0 ;
"CHOLECYSTITIS               "  =   73.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   22.0 ;
"PEPTIC ULCER DISEASE        "  =   61.0
END

RULE 39 [110]
IF
"CHARACTER OF PAIN  "  ==  "COLICKY"
THEN
"APPENDICITIS                "  =   15.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   36.0 ;
"RENAL COLIC                 "  =   52.0 ;
"PERFORATED DUODENAL ULCER   "  =    9.0 ;
"CHOLECYSTITIS               "  =   25.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   80.0 ;
"PEPTIC ULCER DISEASE        "  =   25.0
END
```

```
RULE 40 [120]
IF
"INTENSITY OF PAIN  "  ==  "MODERATE"
THEN
"APPENDICITIS                "  =   63.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   50.0 ;
"RENAL COLIC                 "  =   11.0 ;
"PERFORATED DUODENAL ULCER   "  =    5.0 ;
"CHOLECYSTITIS               "  =   27.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   38.0 ;
"PEPTIC ULCER DISEASE        "  =   44.0
END

RULE 41 [120]
IF
"INTENSITY OF PAIN  "  ==  "SEVERE"
THEN
"APPENDICITIS                "  =   37.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   50.0 ;
"RENAL COLIC                 "  =   89.0 ;
"PERFORATED DUODENAL ULCER   "  =   95.0 ;
"CHOLECYSTITIS               "  =   73.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   62.0 ;
"PEPTIC ULCER DISEASE        "  =   56.0
END

RULE 42 [130]
IF
"AGGRAVATING FACTORS"  ==  "MOVEMENT"
THEN
"APPENDICITIS                "  =   53.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   24.0 ;
"RENAL COLIC                 "  =   17.0 ;
"PERFORATED DUODENAL ULCER   "  =   48.0 ;
"CHOLECYSTITIS               "  =    9.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   18.0 ;
"PEPTIC ULCER DISEASE        "  =   18.0
END

RULE 43 [130]
IF
"AGGRAVATING FACTORS"  ==  "COUGH"
THEN
"APPENDICITIS                "  =   22.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    9.0 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =   12.0 ;
"CHOLECYSTITIS               "  =    6.0 ;
"SMALL BOWEL OBSTRUCTION     "  =    6.0 ;
"PEPTIC ULCER DISEASE        "  =    8.0
END
```

```
RULE 44 [130]
IF
"AGGRAVATING FACTORS"  ==  "BREATHING"
THEN
"APPENDICITIS                "  =    2.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    6.0 ;
"RENAL COLIC                 "  =    3.0 ;
"PERFORATED DUODENAL ULCER   "  =   11.0 ;
"CHOLECYSTITIS               "  =    5.0 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    5.0
END

RULE 45 [130]
IF
"AGGRAVATING FACTORS"  ==  "FOOD"
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    3.0 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =   11.0 ;
"SMALL BOWEL OBSTRUCTION     "  =    2.0 ;
"PEPTIC ULCER DISEASE        "  =    6.0
END

RULE 46 [130]
IF
"AGGRAVATING FACTORS"  ==  "OTHER"
THEN
"APPENDICITIS                "  =    8.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   10.0 ;
"RENAL COLIC                 "  =   10.0 ;
"PERFORATED DUODENAL ULCER   "  =    4.0 ;
"CHOLECYSTITIS               "  =   17.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   14.0 ;
"PEPTIC ULCER DISEASE        "  =   14.0
END

RULE 47 [130]
IF
"AGGRAVATING FACTORS"  ==  "NONE"
THEN
"APPENDICITIS                "  =   22.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   47.0 ;
"RENAL COLIC                 "  =   70.0 ;
"PERFORATED DUODENAL ULCER   "  =   37.0 ;
"CHOLECYSTITIS               "  =   52.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   64.0 ;
"PEPTIC ULCER DISEASE        "  =   55.0
END
```

```
RULE 48 [150]
IF
"PROGRESS OF PAIN    "  ==  "BETTER"
THEN
"APPENDICITIS                 "  =   18.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   39.0 ;
"RENAL COLIC                  "  =   35.0 ;
"PERFORATED DUODENAL ULCER    "  =   10.0 ;
"CHOLECYSTITIS                "  =   18.0 ;
"SMALL BOWEL OBSTRUCTION      "  =   16.0 ;
"PEPTIC ULCER DISEASE         "  =   43.0
END

RULE 49 [150]
IF
"PROGRESS OF PAIN    "  ==  "SAME"
THEN
"APPENDICITIS                 "  =   30.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   35.0 ;
"RENAL COLIC                  "  =   26.0 ;
"PERFORATED DUODENAL ULCER    "  =   44.0 ;
"CHOLECYSTITIS                "  =   49.0 ;
"SMALL BOWEL OBSTRUCTION      "  =   50.0 ;
"PEPTIC ULCER DISEASE         "  =   39.0
END

RULE 50 [150]
IF
"PROGRESS OF PAIN    "  ==  "WORSE"
THEN
"APPENDICITIS                 "  =   52.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   25.0 ;
"RENAL COLIC                  "  =   39.0 ;
"PERFORATED DUODENAL ULCER    "  =   46.0 ;
"CHOLECYSTITIS                "  =   33.0 ;
"SMALL BOWEL OBSTRUCTION      "  =   34.0 ;
"PEPTIC ULCER DISEASE         "  =   18.0
END

RULE 51 [160]
IF
"DURATION OF PAIN    "  <  12
THEN
"APPENDICITIS                 "  =   40.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   68.0 ;
"RENAL COLIC                  "  =   95.0 ;
"PERFORATED DUODENAL ULCER    "  =   87.0 ;
"CHOLECYSTITIS                "  =   71.0 ;
"SMALL BOWEL OBSTRUCTION      "  =   48.0 ;
"PEPTIC ULCER DISEASE         "  =   86.0
END
```

```
RULE 52 [160]
IF
"DURATION OF PAIN    "  >= 12    AND
"DURATION OF PAIN    "  <  24
THEN
"APPENDICITIS                "  =    60.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    32.0 ;
"RENAL COLIC                 "  =     5.0 ;
"PERFORATED DUODENAL ULCER   "  =    13.0 ;
"CHOLECYSTITIS               "  =    29.0 ;
"SMALL BOWEL OBSTRUCTION     "  =    52.0 ;
"PEPTIC ULCER DISEASE        "  =    14.0
END

RULE 53 [160]
IF
"DURATION OF PAIN    "  >= 24    AND
"DURATION OF PAIN    "  <  48
THEN
"APPENDICITIS                "  =     0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =     0.1 ;
"RENAL COLIC                 "  =     0.1 ;
"PERFORATED DUODENAL ULCER   "  =     0.1 ;
"CHOLECYSTITIS               "  =     0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =     0.1 ;
"PEPTIC ULCER DISEASE        "  =     0.1
END

RULE 54 [160]
IF
"DURATION OF PAIN    "  >= 48
THEN
"APPENDICITIS                "  =     0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =     0.1 ;
"RENAL COLIC                 "  =     0.1 ;
"PERFORATED DUODENAL ULCER   "  =     0.1 ;
"CHOLECYSTITIS               "  =     0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =     0.1 ;
"PEPTIC ULCER DISEASE        "  =     0.1
END

RULE 55 [140]
IF
"RELIEVING FACTORS   "  ==  "LYING STILL"
THEN
"APPENDICITIS                "  =    21.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    13.0 ;
"RENAL COLIC                 "  =     6.0 ;
"PERFORATED DUODENAL ULCER   "  =    23.0 ;
"CHOLECYSTITIS               "  =     3.0 ;
"SMALL BOWEL OBSTRUCTION     "  =     2.0 ;
"PEPTIC ULCER DISEASE        "  =    11.0
END
```

```
RULE 56 [140]
IF
"RELIEVING FACTORS  "  ==  "VOMITING"
THEN
"APPENDICITIS                "  =    4.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    6.0 ;
"RENAL COLIC                 "  =    3.0 ;
"PERFORATED DUODENAL ULCER   "  =    2.0 ;
"CHOLECYSTITIS               "  =    2.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   10.0 ;
"PEPTIC ULCER DISEASE        "  =    3.0
END

RULE 57 [140]
IF
"RELIEVING FACTORS  "  ==  "ANTACIDS"
THEN
"APPENDICITIS                "  =    1.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    2.0 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    3.0 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    5.0
END

RULE 58 [140]
IF
"RELIEVING FACTORS  "  ==  "FOOD"
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    1.0 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    1.0 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    3.0
END

RULE 59 [140]
IF
"RELIEVING FACTORS  "  ==  "OTHER"
THEN
"APPENDICITIS                "  =   12.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   12.0 ;
"RENAL COLIC                 "  =   26.0 ;
"PERFORATED DUODENAL ULCER   "  =    6.0 ;
"CHOLECYSTITIS               "  =   21.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   20.0 ;
"PEPTIC ULCER DISEASE        "  =   26.0
END
```

```
RULE 60 [140]
IF
"RELIEVING FACTORS  "  ==  "NONE"
THEN
"APPENDICITIS                "  =   63.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   66.0 ;
"RENAL COLIC                 "  =   65.0 ;
"PERFORATED DUODENAL ULCER   "  =   65.0 ;
"CHOLECYSTITIS               "  =   74.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   68.0 ;
"PEPTIC ULCER DISEASE        "  =   58.0
END

RULE 61 [200]
IF
"NAUSEA              "  ==  "Y"
THEN
"APPENDICITIS                "  =   65.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   62.0 ;
"RENAL COLIC                 "  =   72.0 ;
"PERFORATED DUODENAL ULCER   "  =   61.0 ;
"CHOLECYSTITIS               "  =   68.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   81.0 ;
"PEPTIC ULCER DISEASE        "  =   65.0
END

RULE 62 [200]
IF
"NAUSEA              "  ==  "N"
THEN
"APPENDICITIS                "  =   35.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   38.0 ;
"RENAL COLIC                 "  =   28.0 ;
"PERFORATED DUODENAL ULCER   "  =   39.0 ;
"CHOLECYSTITIS               "  =   32.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   19.0 ;
"PEPTIC ULCER DISEASE        "  =   35.0
END

RULE 63 [210]
IF
"VOMITING            "  ==  "Y"
THEN
"APPENDICITIS                "  =   55.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   42.0 ;
"RENAL COLIC                 "  =   70.0 ;
"PERFORATED DUODENAL ULCER   "  =   54.0 ;
"CHOLECYSTITIS               "  =   72.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   86.0 ;
"PEPTIC ULCER DISEASE        "  =   62.0
END
```

```
RULE 64 [210]
IF
"VOMITING            "  ==  "N"
THEN
"APPENDICITIS                "  =   45.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   58.0 ;
"RENAL COLIC                 "  =   30.0 ;
"PERFORATED DUODENAL ULCER   "  =   46.0 ;
"CHOLECYSTITIS               "  =   28.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   14.0 ;
"PEPTIC ULCER DISEASE        "  =   38.0
END

RULE 65 [220]
IF
"BOWELS              "  ==  "NORMAL"
THEN
"APPENDICITIS                "  =   80.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   86.0 ;
"RENAL COLIC                 "  =   81.0 ;
"PERFORATED DUODENAL ULCER   "  =   87.0 ;
"CHOLECYSTITIS               "  =   75.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   66.0 ;
"PEPTIC ULCER DISEASE        "  =   76.0
END

RULE 66 [220]
IF
"BOWELS              "  ==  "CONSTIPATED"
THEN
"APPENDICITIS                "  =   11.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    4.0 ;
"RENAL COLIC                 "  =   11.0 ;
"PERFORATED DUODENAL ULCER   "  =   11.0 ;
"CHOLECYSTITIS               "  =   16.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   28.0 ;
"PEPTIC ULCER DISEASE        "  =    9.0
END

RULE 67 [220]
IF
"BOWELS              "  ==  "DIARRHEA"
THEN
"APPENDICITIS                "  =    9.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    8.0 ;
"RENAL COLIC                 "  =    8.0 ;
"PERFORATED DUODENAL ULCER   "  =    2.0 ;
"CHOLECYSTITIS               "  =    8.0 ;
"SMALL BOWEL OBSTRUCTION     "  =    6.0 ;
"PEPTIC ULCER DISEASE        "  =   11.0
END
```

```
RULE 68 [220]
IF
"BOWELS                "  ==  "BLOOD IN STOOLS"
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    1.0 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    1.0 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    6.0
END

RULE 69 [220]
IF
"BOWELS                "  ==  "MUCUS IN STOOLS"
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 70 [190]
IF
"APPETITE              "  ==  "Y"
THEN
"APPENDICITIS                "  =   70.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   37.0 ;
"RENAL COLIC                 "  =   40.0 ;
"PERFORATED DUODENAL ULCER   "  =   48.0 ;
"CHOLECYSTITIS               "  =   62.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   66.0 ;
"PEPTIC ULCER DISEASE        "  =   46.0
END

RULE 71 [190]
IF
"APPETITE              "  ==  "N"
THEN
"APPENDICITIS                "  =   30.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   63.0 ;
"RENAL COLIC                 "  =   60.0 ;
"PERFORATED DUODENAL ULCER   "  =   52.0 ;
"CHOLECYSTITIS               "  =   37.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   34.0 ;
"PEPTIC ULCER DISEASE        "  =   54.0
END
```

```
RULE 72 [180]
IF
"JAUNDICE                "  ==  "Y"
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                 "  =    2.0 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    2.0
END

RULE 73 [180]
IF
"JAUNDICE                "  ==  "N"
THEN
"APPENDICITIS                "  =   99.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   99.0 ;
"RENAL COLIC                 "  =   98.0 ;
"PERFORATED DUODENAL ULCER   "  =   99.0 ;
"CHOLECYSTITIS               "  =   99.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   99.0 ;
"PEPTIC ULCER DISEASE        "  =   98.0
END

RULE 74 [230]
IF
"URINE                   "  ==  "NORMAL"
THEN
"APPENDICITIS                "  =   90.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   88.0 ;
"RENAL COLIC                 "  =   59.0 ;
"PERFORATED DUODENAL ULCER   "  =   96.0 ;
"CHOLECYSTITIS               "  =   90.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   94.0 ;
"PEPTIC ULCER DISEASE        "  =   96.0
END

RULE 75 [230]
IF
"URINE                   "  ==  "FREQUENT"
THEN
"APPENDICITIS                "  =    6.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    6.0 ;
"RENAL COLIC                 "  =   23.0 ;
"PERFORATED DUODENAL ULCER   "  =    3.0 ;
"CHOLECYSTITIS               "  =    5.0 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    2.0
END
```

```
RULE 76 [230]
IF
"URINE                "  ==  "PAIN WHEN URINATING"
THEN
"APPENDICITIS                 "  =    4.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    6.0 ;
"RENAL COLIC                  "  =   18.0 ;
"PERFORATED DUODENAL ULCER    "  =    1.0 ;
"CHOLECYSTITIS                "  =    3.0 ;
"SMALL BOWEL OBSTRUCTION      "  =    4.0 ;
"PEPTIC ULCER DISEASE         "  =    0.1
END

RULE 77 [230]
IF
"URINE                "  ==  "DARK (BILE)"
THEN
"APPENDICITIS                 "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                  "  =    4.0 ;
"PERFORATED DUODENAL ULCER    "  =    0.1 ;
"CHOLECYSTITIS                "  =    2.0 ;
"SMALL BOWEL OBSTRUCTION      "  =    2.0 ;
"PEPTIC ULCER DISEASE         "  =    2.0
END

RULE 78 [230]
IF
"URINE                "  ==  "BLOOD IN URINE"
THEN
"APPENDICITIS                 "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                  "  =    1.0 ;
"PERFORATED DUODENAL ULCER    "  =    0.1 ;
"CHOLECYSTITIS                "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION      "  =    0.1 ;
"PEPTIC ULCER DISEASE         "  =    0.1
END

RULE 79 [170]
IF
"INDIGESTION          "  ==  "Y"
THEN
"APPENDICITIS                 "  =   19.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   20.0 ;
"RENAL COLIC                  "  =   18.0 ;
"PERFORATED DUODENAL ULCER    "  =   71.0 ;
"CHOLECYSTITIS                "  =   48.0 ;
"SMALL BOWEL OBSTRUCTION      "  =   21.0 ;
"PEPTIC ULCER DISEASE         "  =   68.0
END
```

```
RULE 80 [170]
IF
"INDIGESTION          "  ==  "N"
THEN
"APPENDICITIS                  "  =   81.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   80.0 ;
"RENAL COLIC                   "  =   82.0 ;
"PERFORATED DUODENAL ULCER     "  =   29.0 ;
"CHOLECYSTITIS                 "  =   52.0 ;
"SMALL BOWEL OBSTRUCTION       "  =   79.0 ;
"PEPTIC ULCER DISEASE          "  =   32.0
END

RULE 81 [250]
IF
"SIMILAR PAIN         "  ==  "Y"
THEN
"APPENDICITIS                  "  =   37.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   40.0 ;
"RENAL COLIC                   "  =   46.0 ;
"PERFORATED DUODENAL ULCER     "  =   47.0 ;
"CHOLECYSTITIS                 "  =   75.0 ;
"SMALL BOWEL OBSTRUCTION       "  =   57.0 ;
"PEPTIC ULCER DISEASE          "  =   67.0
END

RULE 82 [250]
IF
"SIMILAR PAIN         "  ==  "N"
THEN
"APPENDICITIS                  "  =   63.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   60.0 ;
"RENAL COLIC                   "  =   54.0 ;
"PERFORATED DUODENAL ULCER     "  =   53.0 ;
"CHOLECYSTITIS                 "  =   25.0 ;
"SMALL BOWEL OBSTRUCTION       "  =   43.0 ;
"PEPTIC ULCER DISEASE          "  =   33.0
END

RULE 83 [260]
IF
"ABDOMINAL SURGERY    "  ==  "Y"
THEN
"APPENDICITIS                  "  =    2.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   13.0 ;
"RENAL COLIC                   "  =   18.0 ;
"PERFORATED DUODENAL ULCER     "  =   21.0 ;
"CHOLECYSTITIS                 "  =   31.0 ;
"SMALL BOWEL OBSTRUCTION       "  =   86.0 ;
"PEPTIC ULCER DISEASE          "  =   22.0
END
```

```
RULE 84 [260]
IF
"ABDOMINAL SURGERY  "  ==  "N"
THEN
"APPENDICITIS                "  =   98.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   87.0 ;
"RENAL COLIC                 "  =   82.0 ;
"PERFORATED DUODENAL ULCER   "  =   79.0 ;
"CHOLECYSTITIS               "  =   69.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   14.0 ;
"PEPTIC ULCER DISEASE        "  =   78.0
END

RULE 85 [240]
IF
"PREVIOUS ILLNESS   "  ==  "Y"
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 86 [240]
IF
"PREVIOUS ILLNESS   "  ==  "N"
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 87 [270]
IF
"MEDICATION         "  ==  "Y"
THEN
"APPENDICITIS                "  =   10.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   11.0 ;
"RENAL COLIC                 "  =   18.0 ;
"PERFORATED DUODENAL ULCER   "  =   25.0 ;
"CHOLECYSTITIS               "  =   35.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   31.0 ;
"PEPTIC ULCER DISEASE        "  =   44.0
END
```

```
RULE 88 [270]
IF
"MEDICATION          "  ==  "N"
THEN
"APPENDICITIS                "  =   90.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   89.0 ;
"RENAL COLIC                 "  =   82.0 ;
"PERFORATED DUODENAL ULCER   "  =   75.0 ;
"CHOLECYSTITIS               "  =   65.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   69.0 ;
"PEPTIC ULCER DISEASE        "  =   56.0
END

RULE 89 [30]
IF
"TEMPERATURE         "  <  98.6
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 90 [30]
IF
"TEMPERATURE         "  >= 98.6    AND
"TEMPERATURE         "  <= 100.2
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 91 [30]
IF
"TEMPERATURE         "  >= 100.3   AND
"TEMPERATURE         "  <  102
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END
```

```
RULE 92 [30]
IF
"TEMPERATURE          "  >  102
THEN
"APPENDICITIS                  "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                   "  =    0.1 ;
"PERFORATED DUODENAL ULCER     "  =    0.1 ;
"CHOLECYSTITIS                 "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION       "  =    0.1 ;
"PEPTIC ULCER DISEASE          "  =    0.1
END

RULE 93 [40]
IF
"PULSE                "  <  80
THEN
"APPENDICITIS                  "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                   "  =    0.1 ;
"PERFORATED DUODENAL ULCER     "  =    0.1 ;
"CHOLECYSTITIS                 "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION       "  =    0.1 ;
"PEPTIC ULCER DISEASE          "  =    0.1
END

RULE 94 [40]
IF
"PULSE                "  >= 80       AND
"PULSE                "  <= 99
THEN
"APPENDICITIS                  "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                   "  =    0.1 ;
"PERFORATED DUODENAL ULCER     "  =    0.1 ;
"CHOLECYSTITIS                 "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION       "  =    0.1 ;
"PEPTIC ULCER DISEASE          "  =    0.1
END

RULE 95 [40]
IF
"PULSE                "  >  99
THEN
"APPENDICITIS                  "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                   "  =    0.1 ;
"PERFORATED DUODENAL ULCER     "  =    0.1 ;
"CHOLECYSTITIS                 "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION       "  =    0.1 ;
"PEPTIC ULCER DISEASE          "  =    0.1
END
```

```
RULE 96 [60]
IF
"SYSTOLIC            "  <  90
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 97 [60]
IF
"SYSTOLIC            "  >= 90       AND
"SYSTOLIC            "  <= 129
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 98 [60]
IF
"SYSTOLIC            "  >  129
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 99 [70]
IF
"DIASTOLIC           "  <  70
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END
```

```
RULE 100 [70]
IF
"DIASTOLIC            "  >= 70       AND
"DIASTOLIC            "  <= 89
THEN
"APPENDICITIS                  "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                   "  =    0.1 ;
"PERFORATED DUODENAL ULCER     "  =    0.1 ;
"CHOLECYSTITIS                 "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION       "  =    0.1 ;
"PEPTIC ULCER DISEASE          "  =    0.1
END

RULE 101 [70]
IF
"DIASTOLIC            "  >  89
THEN
"APPENDICITIS                  "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                   "  =    0.1 ;
"PERFORATED DUODENAL ULCER     "  =    0.1 ;
"CHOLECYSTITIS                 "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION       "  =    0.1 ;
"PEPTIC ULCER DISEASE          "  =    0.1
END

RULE 102 [80]
IF
"MOOD                 "  ==  "NORMAL"
THEN
"APPENDICITIS                  "  =   71.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   71.0 ;
"RENAL COLIC                   "  =   65.0 ;
"PERFORATED DUODENAL ULCER     "  =   20.0 ;
"CHOLECYSTITIS                 "  =   73.0 ;
"SMALL BOWEL OBSTRUCTION       "  =   38.0 ;
"PEPTIC ULCER DISEASE          "  =   62.0
END

RULE 103 [80]
IF
"MOOD                 "  ==  "DISTRESSED"
THEN
"APPENDICITIS                  "  =   17.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   14.0 ;
"RENAL COLIC                   "  =   28.0 ;
"PERFORATED DUODENAL ULCER     "  =   59.0 ;
"CHOLECYSTITIS                 "  =   14.0 ;
"SMALL BOWEL OBSTRUCTION       "  =   45.0 ;
"PEPTIC ULCER DISEASE          "  =   12.0
END
```

```
RULE 104 [80]
IF
"MOOD                    "  ==  "ANXIOUS"
THEN
"APPENDICITIS                "  =   12.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   15.0 ;
"RENAL COLIC                 "  =    7.0 ;
"PERFORATED DUODENAL ULCER   "  =   20.0 ;
"CHOLECYSTITIS               "  =   14.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   17.0 ;
"PEPTIC ULCER DISEASE        "  =   25.0
END

RULE 105 [280]
IF
"COLOR                   "  ==  "NORMAL"
THEN
"APPENDICITIS                "  =   58.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   81.0 ;
"RENAL COLIC                 "  =   75.0 ;
"PERFORATED DUODENAL ULCER   "  =   43.0 ;
"CHOLECYSTITIS               "  =   69.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   60.0 ;
"PEPTIC ULCER DISEASE        "  =   76.0
END

RULE 106 [280]
IF
"COLOR                   "  ==  "PALE"
THEN
"APPENDICITIS                "  =   14.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   12.0 ;
"RENAL COLIC                 "  =   23.0 ;
"PERFORATED DUODENAL ULCER   "  =   48.0 ;
"CHOLECYSTITIS               "  =   29.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   32.0 ;
"PEPTIC ULCER DISEASE        "  =   16.0
END

RULE 107 [280]
IF
"COLOR                   "  ==  "FLUSHED"
THEN
"APPENDICITIS                "  =   28.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    7.0 ;
"RENAL COLIC                 "  =    2.0 ;
"PERFORATED DUODENAL ULCER   "  =    5.0 ;
"CHOLECYSTITIS               "  =    2.0 ;
"SMALL BOWEL OBSTRUCTION     "  =    5.0 ;
"PEPTIC ULCER DISEASE        "  =    6.0
END
```

```
RULE 108 [280]
IF
"COLOR                "  ==  "JAUNDICED"
THEN
"APPENDICITIS                 "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                  "  =    0.1 ;
"PERFORATED DUODENAL ULCER    "  =    0.1 ;
"CHOLECYSTITIS                "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION      "  =    0.1 ;
"PEPTIC ULCER DISEASE         "  =    2.0
END

RULE 109 [280]
IF
"COLOR                "  ==  "CYANOTIC"
THEN
"APPENDICITIS                 "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                  "  =    0.1 ;
"PERFORATED DUODENAL ULCER    "  =    4.0 ;
"CHOLECYSTITIS                "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION      "  =    3.0 ;
"PEPTIC ULCER DISEASE         "  =    0.1
END

RULE 110 [400]
IF
"WHITE BLOOD COUNT  "  <  8000
THEN
"APPENDICITIS                 "  =    7.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   40.0 ;
"RENAL COLIC                  "  =    1.0 ;
"PERFORATED DUODENAL ULCER    "  =    1.0 ;
"CHOLECYSTITIS                "  =    1.0 ;
"SMALL BOWEL OBSTRUCTION      "  =    1.0 ;
"PEPTIC ULCER DISEASE         "  =   40.0
END

RULE 111 [400]
IF
"WHITE BLOOD COUNT  "  >= 8000    AND
"WHITE BLOOD COUNT  "  <  10000
THEN
"APPENDICITIS                 "  =    7.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   23.0 ;
"RENAL COLIC                  "  =    1.0 ;
"PERFORATED DUODENAL ULCER    "  =    1.0 ;
"CHOLECYSTITIS                "  =    1.0 ;
"SMALL BOWEL OBSTRUCTION      "  =    1.0 ;
"PEPTIC ULCER DISEASE         "  =   23.0
END
```

```
RULE 112 [400]
IF
"WHITE BLOOD COUNT  "  >= 10000   AND
"WHITE BLOOD COUNT  "  <  12000
THEN
"APPENDICITIS                "  =   18.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   17.0 ;
"RENAL COLIC                 "  =    1.0 ;
"PERFORATED DUODENAL ULCER   "  =    1.0 ;
"CHOLECYSTITIS               "  =    1.0 ;
"SMALL BOWEL OBSTRUCTION     "  =    1.0 ;
"PEPTIC ULCER DISEASE        "  =   17.0
END

RULE 113 [400]
IF
"WHITE BLOOD COUNT  "  >= 12000   AND
"WHITE BLOOD COUNT  "  <  15000
THEN
"APPENDICITIS                "  =   32.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   11.0 ;
"RENAL COLIC                 "  =    1.0 ;
"PERFORATED DUODENAL ULCER   "  =    1.0 ;
"CHOLECYSTITIS               "  =    1.0 ;
"SMALL BOWEL OBSTRUCTION     "  =    1.0 ;
"PEPTIC ULCER DISEASE        "  =   11.0
END

RULE 114 [400]
IF
"WHITE BLOOD COUNT  "  >= 15000
THEN
"APPENDICITIS                "  =   35.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    8.0 ;
"RENAL COLIC                 "  =    1.0 ;
"PERFORATED DUODENAL ULCER   "  =    1.0 ;
"CHOLECYSTITIS               "  =    1.0 ;
"SMALL BOWEL OBSTRUCTION     "  =    1.0 ;
"PEPTIC ULCER DISEASE        "  =    8.0
END

RULE 115 [290]
IF
"ABDOMINAL MOVEMENT "  ==  "NORMAL"
THEN
"APPENDICITIS                "  =   87.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   99.0 ;
"RENAL COLIC                 "  =   96.0 ;
"PERFORATED DUODENAL ULCER   "  =   39.0 ;
"CHOLECYSTITIS               "  =   89.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   82.0 ;
"PEPTIC ULCER DISEASE        "  =   92.0
END
```

```
RULE 116 [290]
IF
"ABDOMINAL MOVEMENT "  ==  "PERISTALSIS"
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    6.0 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 117 [290]
IF
"ABDOMINAL MOVEMENT "  ==  "DECREASED"
THEN
"APPENDICITIS                "  =   13.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    1.0 ;
"RENAL COLIC                 "  =    4.0 ;
"PERFORATED DUODENAL ULCER   "  =   61.0 ;
"CHOLECYSTITIS               "  =   11.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   12.0 ;
"PEPTIC ULCER DISEASE        "  =    8.0
END

RULE 118 [300]
IF
"ABDOMINAL SCARS     "  ==  "Y"
THEN
"APPENDICITIS                "  =    2.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   14.0 ;
"RENAL COLIC                 "  =   20.0 ;
"PERFORATED DUODENAL ULCER   "  =   21.0 ;
"CHOLECYSTITIS               "  =   28.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   84.0 ;
"PEPTIC ULCER DISEASE        "  =   23.0
END

RULE 119 [300]
IF
"ABDOMINAL SCARS     "  ==  "N"
THEN
"APPENDICITIS                "  =   98.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   86.0 ;
"RENAL COLIC                 "  =   80.0 ;
"PERFORATED DUODENAL ULCER   "  =   79.0 ;
"CHOLECYSTITIS               "  =   72.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   16.0 ;
"PEPTIC ULCER DISEASE        "  =   77.0
END
```

```
RULE 120 [370]
IF
"GUARDING            "  ==  "Y"
THEN
"APPENDICITIS                 "  =   72.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   24.0 ;
"RENAL COLIC                  "  =   23.0 ;
"PERFORATED DUODENAL ULCER    "  =   62.0 ;
"CHOLECYSTITIS                "  =   62.0 ;
"SMALL BOWEL OBSTRUCTION      "  =   38.0 ;
"PEPTIC ULCER DISEASE         "  =   30.0
END

RULE 121 [370]
IF
"GUARDING            "  ==  "N"
THEN
"APPENDICITIS                 "  =   28.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   76.0 ;
"RENAL COLIC                  "  =   77.0 ;
"PERFORATED DUODENAL ULCER    "  =   38.0 ;
"CHOLECYSTITIS                "  =   38.0 ;
"SMALL BOWEL OBSTRUCTION      "  =   62.0 ;
"PEPTIC ULCER DISEASE         "  =   70.0
END

RULE 122 [380]
IF
"RIGIDITY            "  ==  "Y"
THEN
"APPENDICITIS                 "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                  "  =    0.1 ;
"PERFORATED DUODENAL ULCER    "  =    0.1 ;
"CHOLECYSTITIS                "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION      "  =    0.1 ;
"PEPTIC ULCER DISEASE         "  =    0.1
END

RULE 123 [380]
IF
"RIGIDITY            "  ==  "N"
THEN
"APPENDICITIS                 "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                  "  =    0.1 ;
"PERFORATED DUODENAL ULCER    "  =    0.1 ;
"CHOLECYSTITIS                "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION      "  =    0.1 ;
"PEPTIC ULCER DISEASE         "  =    0.1
END
```

```
RULE 124 [330]
IF
"BOWEL SOUNDS        "  ==  "NORMAL"
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 125 [330]
IF
"BOWEL SOUNDS        "  ==  "ABSENT"
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 126 [330]
IF
"BOWEL SOUNDS        "  ==  "HYPERACTIVE"
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    0.1 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 127 [310]
IF
"DISTENSION          "  ==  "Y"
THEN
"APPENDICITIS                "  =    3.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    1.0 ;
"RENAL COLIC                 "  =    3.0 ;
"PERFORATED DUODENAL ULCER   "  =    3.0 ;
"CHOLECYSTITIS               "  =   10.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   63.0 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END
```

```
RULE 128 [310]
IF
"DISTENSION          "  ==  "N"
THEN
"APPENDICITIS                "  =   97.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   99.0 ;
"RENAL COLIC                 "  =   97.0 ;
"PERFORATED DUODENAL ULCER   "  =   97.0 ;
"CHOLECYSTITIS               "  =   90.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   37.0 ;
"PEPTIC ULCER DISEASE        "  =   99.0
END

RULE 129 [320]
IF
"SWELLING            "  ==  "Y"
THEN
"APPENDICITIS                "  =    1.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    1.0 ;
"RENAL COLIC                 "  =    5.0 ;
"PERFORATED DUODENAL ULCER   "  =    1.0 ;
"CHOLECYSTITIS               "  =   11.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   12.0 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 130 [320]
IF
"SWELLING            "  ==  "N"
THEN
"APPENDICITIS                "  =   99.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   99.0 ;
"RENAL COLIC                 "  =   95.0 ;
"PERFORATED DUODENAL ULCER   "  =   99.0 ;
"CHOLECYSTITIS               "  =   89.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   88.0 ;
"PEPTIC ULCER DISEASE        "  =   99.0
END

RULE 131 [340]
IF
"TENDERNESS          "  ==  "RIGHT UPPER QUAD"
THEN
"APPENDICITIS                "  =    1.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    4.0 ;
"RENAL COLIC                 "  =    1.0 ;
"PERFORATED DUODENAL ULCER   "  =    3.0 ;
"CHOLECYSTITIS               "  =   69.0 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =   11.0
END
```

```
RULE 132 [340]
IF
"TENDERNESS          "  ==  "LEFT UPPER QUAD"
THEN
"APPENDICITIS                  "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    5.0 ;
"RENAL COLIC                   "  =    1.0 ;
"PERFORATED DUODENAL ULCER     "  =    0.1 ;
"CHOLECYSTITIS                 "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION       "  =    0.1 ;
"PEPTIC ULCER DISEASE          "  =    3.0
END

RULE 133 [340]
IF
"TENDERNESS          "  ==  "RIGHT LOWER QUAD"
THEN
"APPENDICITIS                  "  =   87.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   29.0 ;
"RENAL COLIC                   "  =   15.0 ;
"PERFORATED DUODENAL ULCER     "  =    3.0 ;
"CHOLECYSTITIS                 "  =    2.0 ;
"SMALL BOWEL OBSTRUCTION       "  =    6.0 ;
"PEPTIC ULCER DISEASE          "  =    2.0
END

RULE 134 [340]
IF
"TENDERNESS          "  ==  "LEFT LOWER QUAD"
THEN
"APPENDICITIS                  "  =    2.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   11.0 ;
"RENAL COLIC                   "  =   12.0 ;
"PERFORATED DUODENAL ULCER     "  =    1.0 ;
"CHOLECYSTITIS                 "  =    2.0 ;
"SMALL BOWEL OBSTRUCTION       "  =    6.0 ;
"PEPTIC ULCER DISEASE          "  =    0.1
END

RULE 135 [340]
IF
"TENDERNESS          "  ==  "UPPER HALF"
THEN
"APPENDICITIS                  "  =    2.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   11.0 ;
"RENAL COLIC                   "  =    0.1 ;
"PERFORATED DUODENAL ULCER     "  =   30.0 ;
"CHOLECYSTITIS                 "  =   14.0 ;
"SMALL BOWEL OBSTRUCTION       "  =   14.0 ;
"PEPTIC ULCER DISEASE          "  =   61.0
END
```

```
RULE 136 [340]
IF
"TENDERNESS          "  ==  "LOWER HALF"
THEN
"APPENDICITIS                  "  =    2.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   11.0 ;
"RENAL COLIC                   "  =    2.0 ;
"PERFORATED DUODENAL ULCER     "  =    1.0 ;
"CHOLECYSTITIS                 "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION       "  =   10.0 ;
"PEPTIC ULCER DISEASE          "  =    3.0
END

RULE 137 [340]
IF
"TENDERNESS          "  ==  "RIGHT HALF"
THEN
"APPENDICITIS                  "  =    7.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    6.0 ;
"RENAL COLIC                   "  =   11.0 ;
"PERFORATED DUODENAL ULCER     "  =   10.0 ;
"CHOLECYSTITIS                 "  =    3.0 ;
"SMALL BOWEL OBSTRUCTION       "  =    2.0 ;
"PEPTIC ULCER DISEASE          "  =    6.0
END

RULE 138 [340]
IF
"TENDERNESS          "  ==  "LEFT HALF"
THEN
"APPENDICITIS                  "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    5.0 ;
"RENAL COLIC                   "  =   11.0 ;
"PERFORATED DUODENAL ULCER     "  =    0.1 ;
"CHOLECYSTITIS                 "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION       "  =    2.0 ;
"PEPTIC ULCER DISEASE          "  =    2.0
END

RULE 139 [340]
IF
"TENDERNESS          "  ==  "CENTRAL"
THEN
"APPENDICITIS                  "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    3.0 ;
"RENAL COLIC                   "  =    0.1 ;
"PERFORATED DUODENAL ULCER     "  =    1.0 ;
"CHOLECYSTITIS                 "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION       "  =   14.0 ;
"PEPTIC ULCER DISEASE          "  =    2.0
END
```

```
RULE 140 [340]
IF
"TENDERNESS          "  ==  "GENERAL"
THEN
"APPENDICITIS                "  =    3.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    1.0 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =   55.0 ;
"CHOLECYSTITIS               "  =    2.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   40.0 ;
"PEPTIC ULCER DISEASE        "  =    8.0
END

RULE 141 [340]
IF
"TENDERNESS          "  ==  "RIGHT FLANK"
THEN
"APPENDICITIS                "  =    1.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    1.0 ;
"RENAL COLIC                 "  =   19.0 ;
"PERFORATED DUODENAL ULCER   "  =    1.0 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 142 [340]
IF
"TENDERNESS          "  ==  "LEFT FLANK"
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    1.0 ;
"RENAL COLIC                 "  =   23.0 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    0.1
END

RULE 143 [340]
IF
"TENDERNESS          "  ==  "NONE"
THEN
"APPENDICITIS                "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   19.0 ;
"RENAL COLIC                 "  =   19.0 ;
"PERFORATED DUODENAL ULCER   "  =    0.1 ;
"CHOLECYSTITIS               "  =   11.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   10.0 ;
"PEPTIC ULCER DISEASE        "  =   12.0
END
```

```
RULE 144 [360]
IF
"MURPHY'S SIGN      "  ==  "Y"
THEN
"APPENDICITIS                "  =    1.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    3.0 ;
"RENAL COLIC                 "  =    0.1 ;
"PERFORATED DUODENAL ULCER   "  =    3.0 ;
"CHOLECYSTITIS               "  =   68.0 ;
"SMALL BOWEL OBSTRUCTION     "  =    0.1 ;
"PEPTIC ULCER DISEASE        "  =    2.0
END

RULE 145 [360]
IF
"MURPHY'S SIGN      "  ==  "N"
THEN
"APPENDICITIS                "  =   99.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   97.0 ;
"RENAL COLIC                 "  =   99.0 ;
"PERFORATED DUODENAL ULCER   "  =   97.0 ;
"CHOLECYSTITIS               "  =   32.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   99.0 ;
"PEPTIC ULCER DISEASE        "  =   98.0
END

RULE 146 [350]
IF
"REBOUND TENDERNESS "  ==  "Y"
THEN
"APPENDICITIS                "  =   80.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   23.0 ;
"RENAL COLIC                 "  =    5.0 ;
"PERFORATED DUODENAL ULCER   "  =   54.0 ;
"CHOLECYSTITIS               "  =   10.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   33.0 ;
"PEPTIC ULCER DISEASE        "  =   18.0
END

RULE 147 [350]
IF
"REBOUND TENDERNESS "  ==  "N"
THEN
"APPENDICITIS                "  =   20.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   77.0 ;
"RENAL COLIC                 "  =   95.0 ;
"PERFORATED DUODENAL ULCER   "  =   46.0 ;
"CHOLECYSTITIS               "  =   90.0 ;
"SMALL BOWEL OBSTRUCTION     "  =   66.0 ;
"PEPTIC ULCER DISEASE        "  =   82.0
END
```

```
RULE 148 [390]
IF
"RECTAL EXAM          "  ==  "NORMAL"
THEN
"APPENDICITIS                  "  =   57.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   72.0 ;
"RENAL COLIC                   "  =   91.0 ;
"PERFORATED DUODENAL ULCER     "  =   80.0 ;
"CHOLECYSTITIS                 "  =   92.0 ;
"SMALL BOWEL OBSTRUCTION       "  =   91.0 ;
"PEPTIC ULCER DISEASE          "  =   98.0
END

RULE 149 [390]
IF
"RECTAL EXAM          "  ==  "SHOWS MASS"
THEN
"APPENDICITIS                  "  =    0.1 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    1.0 ;
"RENAL COLIC                   "  =    0.1 ;
"PERFORATED DUODENAL ULCER     "  =    0.1 ;
"CHOLECYSTITIS                 "  =    2.0 ;
"SMALL BOWEL OBSTRUCTION       "  =    0.1 ;
"PEPTIC ULCER DISEASE          "  =    0.1
END

RULE 150 [390]
IF
"RECTAL EXAM          "  ==  "LEFT RECTAL TENDERNESS"
THEN
"APPENDICITIS                  "  =    3.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    4.0 ;
"RENAL COLIC                   "  =    5.0 ;
"PERFORATED DUODENAL ULCER     "  =    0.1 ;
"CHOLECYSTITIS                 "  =    0.1 ;
"SMALL BOWEL OBSTRUCTION       "  =    0.1 ;
"PEPTIC ULCER DISEASE          "  =    0.1
END

RULE 151 [390]
IF
"RECTAL EXAM          "  ==  "RIGHT RECTAL TENDERNESS"
THEN
"APPENDICITIS                  "  =   27.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =   14.0 ;
"RENAL COLIC                   "  =    0.1 ;
"PERFORATED DUODENAL ULCER     "  =    3.0 ;
"CHOLECYSTITIS                 "  =    2.0 ;
"SMALL BOWEL OBSTRUCTION       "  =    3.0 ;
"PEPTIC ULCER DISEASE          "  =    2.0
END
```

```
RULE 152 [390]
IF
"RECTAL EXAM          "  ==  "GENERALIZED"
THEN
"APPENDICITIS                  "  =   12.0 ;
"NON SPECIFIC ABDOMINAL PAIN"  =    9.0 ;
"RENAL COLIC                   "  =    4.0 ;
"PERFORATED DUODENAL ULCER     "  =   17.0 ;
"CHOLECYSTITIS                 "  =    4.0 ;
"SMALL BOWEL OBSTRUCTION       "  =    6.0 ;
"PEPTIC ULCER DISEASE          "  =    0.1
END

RULE 153 [10]
IF
"APRIORI" != "Y"
THEN
"APPENDICITIS                  "  =  18  ;
"NON SPECIFIC ABDOMINAL PAIN"  =  54  ;
"RENAL COLIC                   "  =  3   ;
"PERFORATED DUODENAL ULCER     "  =  0.1 ;
"CHOLECYSTITIS                 "  =  5   ;
"SMALL BOWEL OBSTRUCTION       "  =  3   ;
"PEPTIC ULCER DISEASE          "  =  16
END
```

APPENDIX D. CAMD EXPERT SYSTEM SOFTWARE TEST PLAN

CONTENTS

1. INTRODUCTION

This document describes the CAMD expert system software test plan. There will be two basic test procedures for the CAMD module, namely, an interactive test and a batch test. The purpose of the interactive system test is to verify that all components of the system are functioning correctly. This is not a medical validation. It is a test only of the software components. The purpose of the batch test is to medically validate the system. In order to to this, the batch test procedure must be followed with sufficient test cases to prove the validity of the system's diagnostic capability.

The batch test runs directly off of a test driver file, and provided that no anomalies are discovered in the test file, requires no data to be input by the user. Instead, the complete test script and all stimuli are supplied to the CAMD module from the test file. The CAMD expert system software will log the test results to a test log file for off-line recording and analysis.

2. TEST FILE FORMATS

The following is a provisional specification of the test file formats.

2.1 Test Transaction Directory

The test will be run off of a set of test transaction files, stored in an off-line test subdirectory. This directory will consist of the following files, all pre-initialized and pre-indexed in the FoxPro 2.0 environment:

ENCOUNTR.DBF
SYMPTOMS.DBF
DIAGNOSE.DBF

If these transaction files contain any encounters, they can be used in the test run or replaced with new cases from the test driver file. All test results are written to the transaction files and can be retrieved using the test case id.

2.2 TEST.DBF Driver File

The test file is a FoxPro 2.0 compatible .DBF file in which each separate test is entered as a record.

2.2.1 Test Record Format

The format of a record is as follows:

NAME	TYPE	WIDTH	DESCRIPTION
CASEID	N	5	ID number for the test case
AREA	N	5	complaint area number
HASDATA	L	1	true if CASEDATA field is valid
CASEDATA	C	254	string of logical 1s and 0s for symptoms
OUTCOME	C	254	<name>=<confidence> pairs,comma delimited.

2.2.2 Description of Fields

CASEID

ID number for the test case. This number is the same as the ENCOUNTER ID in the on-line transaction file for CAMD, ENCOUNTR.DBF. For test purposes, the ID can be any unique number to identify the test cases. The caseid is used to enter test data into the test transaction files described in the previous section.

AREA

Complaint area number. This number must be one of the defined complaint areas handled by the CAMD expert system. For the current series of tests, only the following codes can be used:

1 = abdomen
2 = chest

HASDATA

Logical true if CASEDATA field has valid data; otherwise, it must be false or blank. If HASDATA is true, then the test case will be read from the CASEDATA field. If HASDATA is false, then the test case is read from the ENCOUNTR.DBF file described above.

OUTCOME

Character string describing diagnostic outcome. Consists of: <name>=<confidence factor> pairs, comma delimited.

CASEDATA

Character string with logical 1 for each true condition, logical 0 for each false condition, in the order described below.

2.2.3 Abdominal Pain Case

Bit	Condition
1	"SEX " == "M"
2	"SEX " == "F"
3	"AGE " >= 0 AND "AGE " < 10
4	"AGE " >= 10 AND "AGE " < 20
5	"AGE " >= 20 AND "AGE " < 30
6	"AGE " >= 30 AND "AGE " < 40
7	"AGE " >= 40 AND "AGE " < 50
8	"AGE " >= 50 AND "AGE " < 60
9	"AGE " >= 60 AND "AGE " < 70
10	"AGE " >= 70
11	"SITE AT ONSET " == "RIGHT UPPER QUAD"
12	"SITE AT ONSET " == "UPPER QUAD"
13	"SITE AT ONSET " == "RIGHT LOWER QUAD"
14	"SITE AT ONSET " == "LEFT LOWER QUAD"
15	"SITE AT ONSET " == "UPPER HALF"
16	"SITE AT ONSET " == "LOWER HALF"
17	"SITE AT ONSET " == "RIGHT HALF"
18	"SITE AT ONSET " == "LEFT HALF"
19	"SITE AT ONSET " == "CENTRAL"

```
20  "SITE AT ONSET      "  ==  "GENERAL"
21  "SITE AT ONSET      "  ==  "RIGHT FLANK"
22  "SITE AT ONSET      "  ==  "LEFT FLANK"
23  "SITE AT ONSET      "  ==  "NO PAIN AT ONSET"
24  "SITE AT PRESENT    "  ==  "RIGHT UPPER QUAD"
25  "SITE AT PRESENT    "  ==  "LEFT UPPER QUAD"
26  "SITE AT PRESENT    "  ==  "RIGHT LOWER QUAD"
27  "SITE AT PRESENT    "  ==  "LEFT LOWER QUAD"
28  "SITE AT PRESENT    "  ==  "UPPER HALF"
29  "SITE AT PRESENT    "  ==  "LOWER HALF"
30  "SITE AT PRESENT    "  ==  "RIGHT HALF"
31  "SITE AT PRESENT    "  ==  "LEFT HALF"
32  "SITE AT PRESENT    "  ==  "CENTRAL"
33  "SITE AT PRESENT    "  ==  "GENERAL"
34  "SITE AT PRESENT    "  ==  "RIGHT FLANK"
35  "SITE AT PRESENT    "  ==  "LEFT FLANK"
36  "SITE AT PRESENT    "  ==  "NO PAIN AT PRESENT"
37  "CHARACTER OF PAIN  "  ==  "INTERMITTENT"
38  "CHARACTER OF PAIN  "  ==  "STEADY"
39  "CHARACTER OF PAIN  "  ==  "COLICKY"
40  "INTENSITY OF PAIN  "  ==  "MODERATE"
41  "INTENSITY OF PAIN  "  ==  "SEVERE"
42  "AGGRAVATING FACTORS"  ==  "MOVEMENT"
43  "AGGRAVATING FACTORS"  ==  "COUGH"
44  "AGGRAVATING FACTORS"  ==  "BREATHING"
45  "AGGRAVATING FACTORS"  ==  "FOOD"
46  "AGGRAVATING FACTORS"  ==  "OTHER"
47  "AGGRAVATING FACTORS"  ==  "NONE"
48  "PROGRESS OF PAIN   "  ==  "BETTER"
49  "PROGRESS OF PAIN   "  ==  "SAME"
50  "PROGRESS OF PAIN   "  ==  "WORSE"
51  "DURATION OF PAIN   "  <   12
52  "DURATION OF PAIN   "  >=  12   AND "DURATION OF PAIN   "  <   24
53  "DURATION OF PAIN   "  >=  24   AND "DURATION OF PAIN   "  <   48
54  "DURATION OF PAIN   "  >=  48
55  "RELIEVING FACTORS  "  ==   "LYING STILL"
56  "RELIEVING FACTORS  "  ==   "VOMITING"
57  "RELIEVING FACTORS  "  ==   "ANTACIDS"
58  "RELIEVING FACTORS  "  ==   "FOOD"
59  "RELIEVING FACTORS  "  ==   "OTHER"
60  "RELIEVING FACTORS  "  ==   "NONE"
61  "NAUSEA             "  ==   "Y"
62  "NAUSEA             "  ==   "N"
63  "VOMITING           "  ==   "Y"
64  "VOMITING           "  ==   "N"
65  "BOWELS             "  ==   "NORMAL"
66  "BOWELS             "  ==   "CONSTIPATED"
67  "BOWELS             "  ==   "DIARRHEA"
68  "BOWELS             "  ==   "BLOOD IN STOOLS"
69  "BOWELS             "  ==   "MUCUS IN STOOLS"
70  "APPETITE           "  ==   "Y"
71  "APPETITE           "  ==   "N"
```

```
72  "JAUNDICE              "  ==  "Y"
73  "JAUNDICE              "  ==  "N"
74  "URINE                 "  ==  "NORMAL"
75  "URINE                 "  ==  "FREQUENT"
76  "URINE                 "  ==  "PAIN WHEN URINATING"
77  "URINE                 "  ==  "DARK (BILE)"
78  "URINE                 "  ==  "BLOOD IN URINE"
79  "INDIGESTION           "  ==  "Y"
80  "INDIGESTION           "  ==  "N"
81  "SIMILAR PAIN          "  ==  "Y"
82  "SIMILAR PAIN          "  ==  "N"
83  "ABDOMINAL SURGERY     "  ==  "Y"
84  "ABDOMINAL SURGERY     "  ==  "N"
85  "PREVIOUS ILLNESS      "  ==  "Y"
86  "PREVIOUS ILLNESS      "  ==  "N"
87  "MEDICATION            "  ==  "Y"
88  "MEDICATION            "  ==  "N"
89  "TEMPERATURE           "  <   98.6
90  "TEMPERATURE           "  >=  98.6    AND "TEMPERATURE    "  < 100.3
91  "TEMPERATURE           "  >=  100.3   AND "TEMPERATURE    "  <= 102
92  "TEMPERATURE           "  >   102
93  "PULSE                 "  <   80
94  "PULSE                 "  >=  80      AND "PULSE          "  <= 99
95  "PULSE                 "  >   99
96  "SYSTOLIC              "  <   90
97  "SYSTOLIC              "  >=  90      AND "SYSTOLIC       "  <= 129
98  "SYSTOLIC              "  >   129
99  "DIASTOLIC             "  <   70
100 "DIASTOLIC             "  >=  70      AND "DIASTOLIC      "  <= 89
101 "DIASTOLIC             "  >   89
102 "MOOD                  "  ==  "NORMAL"
103 "MOOD                  "  ==  "DISTRESSED"
104 "MOOD                  "  ==  "ANXIOUS"
105 "COLOR                 "  ==  "NORMAL"
106 "COLOR                 "  ==  "PALE"
107 "COLOR                 "  ==  "FLUSHED"
108 "COLOR                 "  ==  "JAUNDICED"
109 "COLOR                 "  ==  "CYANOTIC"
110 "WHITE BLOOD COUNT     "  <   8000
111 "WHITE BLOOD COUNT     "  >=  8000 AND "WHITE BLOOD COUNT" < 10000
112 "WHITE BLOOD COUNT     "  >=  10000 AND "WHITE BLOOD COUNT" < 12000
113 "WHITE BLOOD COUNT     "  >=  12000 AND "WHITE BLOOD COUNT" < 15000
114 "WHITE BLOOD COUNT     "  >=  15000
115 "ABDOMINAL MOVEMENT    "  ==  "NORMAL"
116 "ABDOMINAL MOVEMENT    "  ==  "PERISTALSIS"
117 "ABDOMINAL MOVEMENT    "  ==  "DECREASED"
118 "ABDOMINAL SCARS       "  ==  "Y"
119 "ABDOMINAL SCARS       "  ==  "N"
120 "GUARDING              "  ==  "Y"
121 "GUARDING              "  ==  "N"
122 "RIGIDITY              "  ==  "Y"
123 "RIGIDITY              "  ==  "N"
```

```
124 "BOWEL SOUNDS          "  ==  "NORMAL"
125 "BOWEL SOUNDS          "  ==  "ABSENT"
126 "BOWEL SOUNDS          "  ==  "HYPERACTIVE"
127 "DISTENSION            "  ==  "Y"
128 "DISTENSION            "  ==  "N"
129 "SWELLING              "  ==  "Y"
130 "SWELLING              "  ==  "N"
131 "TENDERNESS            "  ==  "RIGHT UPPER QUAD"
132 "TENDERNESS            "  ==  "LEFT UPPER QUAD"
133 "TENDERNESS            "  ==  "RIGHT LOWER QUAD"
134 "TENDERNESS            "  ==  "LEFT LOWER QUAD"
135 "TENDERNESS            "  ==  "UPPER HALF"
136 "TENDERNESS            "  ==  "LOWER HALF"
137 "TENDERNESS            "  ==  "RIGHT HALF"
138 "TENDERNESS            "  ==  "LEFT HALF"
139 "TENDERNESS            "  ==  "CENTRAL"
140 "TENDERNESS            "  ==  "GENERAL"
141 "TENDERNESS            "  ==  "RIGHT FLANK"
142 "TENDERNESS            "  ==  "LEFT FLANK"
143 "TENDERNESS            "  ==  "NONE"
144 "MURPHY'S SIGN         "  ==  "Y"
145 "MURPHY'S SIGN         "  ==  "N"
146 "REBOUND TENDERNESS    "  ==  "Y"
147 "REBOUND TENDERNESS    "  ==  "N"
148 "RECTAL EXAM           "  ==  "NORMAL"
149 "RECTAL EXAM           "  ==  "SHOWS MASS"
150 "RECTAL EXAM           "  ==  "LEFT RECTAL TENDERNESS"
151 "RECTAL EXAM           "  ==  "RIGHT RECTAL TENDERNESS"
152 "RECTAL EXAM           "  ==  "GENERALIZED"
153 "APRIORI               "  !=  "Y"
```

2.2.4 Chest Pain Case

```
Bit Condition
--- ------------------------------------------------------------
1   "SEX"  == "M"
2   "SEX" == "F"
3   "AGE" >= 17 AND "AGE" < 30
4   "AGE" >= 30 AND "AGE" < 40
5   "AGE" >= 40 AND "AGE" < 50
6   "AGE" >= 50 AND "AGE" < 60
7   "AGE" >= 60 AND "AGE" < 70
8   "AGE" > 70
9   "SITE OF PAIN"  == "CENTRAL"
10  "SITE OF PAIN"  == "CHEST"
11  "SITE OF PAIN"  == "ACROSS"
12  "SITE OF PAIN"  == "L SIDE"
13  "SITE OF PAIN"  == "R SIDE"
14  "SITE OF PAIN"  == "EPIGASTRIC"
15  "SITE OF PAIN"  == "OTHER"
16  "RADIATION"  == "YES"
```

```
17  "RADIATION"  == "NONE"
18  "RADIATION"  == "L ARM"
19  "RADIATION"  == "R ARM"
20  "RADIATION"  == "BOTH ARMS"
21  "RADIATION"  == "BACK"
22  "RADIATION"  == "CHEST"
23  "RADIATION"  == "SHOULDERS"
24  "RADIATION"  == "NECK"
25  "RADIATION"  == "JAW"
26  "RADIATION"  == "THROAT"
27  "RADIATION"  == "FINGERS HANDS"
28  "RADIATION"  == "EPIGASTRIC"
29  "RADIATION"  == "OTHER"
30  "DURATION OF PAIN" < 1
31  "DURATION OF PAIN" >= 1  AND "DURATION OF PAIN" < 2
32  "DURATION OF PAIN" >= 2  AND "DURATION OF PAIN" < 4
33  "DURATION OF PAIN" >= 4  AND "DURATION OF PAIN" < 12
34  "DURATION OF PAIN" >= 12 AND "DURATION OF PAIN" < 24
35  "DURATION OF PAIN" >= 24 AND "DURATION OF PAIN" < 168
36  "DURATION OF PAIN" >= 168
37  "ONSET OF PAIN"  == "SUDDEN"
38  "ONSET OF PAIN"  == "GRADUAL"
39  "TIME COURSE OF PAIN" == "CONTINUOUS"
40  "TIME COURSE OF PAIN" == "INTERMITTENT"
41  "TYPE OF PAIN" == "TIGHT"
42  "TYPE OF PAIN" == "SHARP"
43  "TYPE OF PAIN" == "HEAVY PRESSING CRUSH"
44  "TYPE OF PAIN" == "GRIPPING"
45  "TYPE OF PAIN" == "BURNING"
46  "TYPE OF PAIN" == "ACHING"
47  "TYPE OF PAIN" == "DULL"
48  "TYPE OF PAIN" == "STABBING"
49  "TYPE OF PAIN" == "NAGGING"
50  "NUMBNESS" == "Y"
51  "NUMBNESS" == "N"
52  "SEVERITY" == "MODERATE"
53  "SEVERITY" == "SEVERE"
54  "AGGRAVATING FACTORSC" == "MOVEMENT"
55  "AGGRAVATING FACTORSC" == "COUGH"
56  "AGGRAVATING FACTORSC" == "BREATHING"
57  "AGGRAVATING FACTORSC" == "SITTING"
58  "AGGRAVATING FACTORSC" == "LYING DOWN REST"
59  "AGGRAVATING FACTORSC" == "LEANING FORWARD"
60  "AGGRAVATING FACTORSC" == "OTHER"
61  "AGGRAVATING FACTORSC" == "NONE"
62  "PROGRESS OF PAIN" == "BETTER"
63  "PROGRESS OF PAIN" == "SAME"
64  "PROGRESS OF PAIN" == "WORSE"
65  "RELIEVING FACTORS C" == "NITRO"
66  "RELIEVING FACTORS C" == "REST"
67  "RELIEVING FACTORS C" == "WALKING"
68  "RELIEVING FACTORS C" == "MORPHINE"
```

```
69  "RELIEVING FACTORS C" == "OTHER DRUGS"
70  "RELIEVING FACTORS C" == "OTHER"
71  "RELIEVING FACTORS C" == "NONE"
72  "DYSPNEA" == "ABSENT"
73  "DYSPNEA" == "THIS ILLNESS"
74  "DYSPNEA" == "CHRONIC"
75  "COUGH" == "ABSENT"
76  "COUGH" == "THIS ILLNESS"
77  "COUGH" == "CHRONIC"
78  "SPUTUM" == "Y"
79  "SPUTUM" == "N"
80  "ORTHOPNEA" == "Y"
81  "ORTHOPNEA" == "N"
82  "PND" == "Y"
83  "PND" == "N"
84  "REFLUX" == "Y"
85  "REFLUX" == "N"
86  "NAUSEA" == "Y"
87  "NAUSEA" == "N"
88  "VOMITING" == "Y"
89  "VOMITING" == "N"
90  "APPETITE" == "Y"
91  "APPETITE" == "N"
92  "BOWEL HABITS" == "NORMAL"
93  "BOWEL HABITS" == "CONSTIPATED"
94  "BOWEL HABITS" == "DIARRHEA"
95  "PREV CHEST PAIN" == "Y"
96  "PREV CHEST PAIN" == "N"
97  "PREV CR ILLNESS" == "Y"
98  "PREV CR ILLNESS" == "N"
99  "PREV MAJOR SURGERY" == "Y"
100 "PREV MAJOR SURGERY" == "N"
101 "SMOKER" == "Y"
102 "SMOKER" == "N"
103 "POSITIVE HISTORY" == "MI"
104 "POSITIVE HISTORY" == "ANGINA"
105 "POSITIVE HISTORY" == "BRONCHITIS"
106 "POSITIVE HISTORY" == "HYPERTENSION"
107 "POSITIVE HISTORY" == "DIABETES"
108 "TEMPERATURE" >= 97.8 AND "TEMPERATURE" <= 99.6
109 "TEMPERATURE" > 99.6
110 "TEMPERATURE" < 97.8
111 "PULSE" <= 60
112 "PULSE" > 60 AND "PULSE" <= 70
113 "PULSE" > 70 AND "PULSE" <= 80
114 "PULSE" > 80 AND "PULSE" <= 100
115 "PULSE" > 100
116 "RESPIRATION" < 20
117 "RESPIRATION" == 20
118 "RESPIRATION" > 20 AND "RESPIRATION" <= 25
119 "RESPIRATION" > 25 AND "RESPIRATION" <= 30
120 "RESPIRATION" > 30
```

```
121 "SYSTOLIC" <= 100
122 "SYSTOLIC" > 100 AND "SYSTOLIC" <= 120
123 "SYSTOLIC" > 120 AND "SYSTOLIC" <= 140
124 "SYSTOLIC" > 140 AND "SYSTOLIC" <= 160
125 "SYSTOLIC" > 160
126 "DIASTOLIC" <= 70
127 "DIASTOLIC" > 70 AND "DIASTOLIC" <= 80
128 "DIASTOLIC" > 80 AND "DIASTOLIC" <= 90
129 "DIASTOLIC" > 90 AND "DIASTOLIC" <= 100
130 "DIASTOLIC" > 100
131 "ECG" == "ST ELEVATION"
132 "ECG" == "T DEPRESSION"
133 "ECG" == "Q WAVES"
134 "ECG" == "ST DEPRESSION"
135 "ECG" == "ARRYTHMIA"
136 "ECG" == "NORMAL"
137 "SGOT" < 25
138 "SGOT" >= 25  AND "SGOT" <= 50
139 "SGOT" > 50  AND "SGOT" <= 100
140 "SGOT" > 100 AND "SGOT" <= 200
141 "SGOT" > 200
142 "MOOD C" == "NORMAL"
143 "MOOD C" == "ANXIOUS"
144 "MOOD C" == "DISTRESSED"
145 "MOOD C" == "IN SHOCK"
146 "COLOR C" == "NORMAL"
147 "COLOR C" == "PALE"
148 "COLOR C" == "FLUSHED"
149 "COLOR C" == "CYANOTIC"
150 "EDEMA" == "ABSENT"
151 "EDEMA" == "ANKLES"
152 "EDEMA" == "OTHER"
153 "SWEATING" == "Y"
154 "SWEATING" == "N"
155 "SHIVERING" == "Y"
156 "SHIVERING" == "N"
157 "RESPIRATORY MOVEMENT" == "NORMAL"
158 "RESPIRATORY MOVEMENT" == "ABNORMAL"
159 "PERCUSSION" == "NORMAL"
160 "PERCUSSION" == "DULL"
161 "PERCUSSION" == "HYPERRESONANT"
162 "CHEST SOUNDS" == "NORMAL"
163 "CHEST SOUNDS" == "RHONCHI"
164 "CHEST SOUNDS" == "RALES"
165 "CHEST SOUNDS" == "DECREASED"
166 "COLD CLAMMY" == "Y"
167 "COLD CLAMMY" == "N"
168 "CALF TENDERNESS" == "Y"
169 "CALF TENDERNESS" == "N"
170 "CHEST WALL TENDER" == "Y"
171 "CHEST WALL TENDER" == "N"
172 "JUGULAR VEN PRESSURE" == "NORMAL"
```

```
173 "JUGULAR VEN PRESSURE" == "RAISED"
174 "JUGULAR VEN PRESSURE" == "LOWERED"
175 "HEART SOUNDS" == "NORMAL"
176 "HEART SOUNDS" == "ABNORMAL"
177 "APRIORI" != "Y"
```

2.3 Test Log File

This is an optional file that can be used to log the test procedure as it progresses. The log file will contain the following information as free format text:

Name of test program.
Version number of test program.
Organizational credits and copyright notices.
Date and time of test.
Result of each test case including:

- case id
- outcome
- case data

Test summary including:

- Number of test cases read.
- Number of test cases successfully completed.

This information is also displayed on the screen during the test.

3. INTERACTIVE TEST PROCEDURE

This section describes the procedures for running and evaluating the interactive test. The purpose of the interactive test is to assure that the required software components have been implemented correctly. The following major components must be checked out during the test:

CAMD sign-on screen
User logon
CAMD main menu
Register Patient
Patient encounter
Help
On-line medical reference

List of screens:

2 print encounter
3 patients on file
4 patient details

5 complaints
6 edit number
6a edit check box
6b edit radio button
6c edit logical
7 diseases on file
8 view disease
9 warning
10 confirm
11 pending encounters
12 patient encounter

Detailed test procedures follow:

---

1 CAMD Startup
This test assures that the system can be started properly.

1. Start CAMD program
   The program is started from MS-DOS command line
2. Displays system menu bar and signon screen

---

2 CAMD Shutdown
This test assures that the system can be shut down properly.

1. Start CAMD (see Test 1)
3. File (Alt F or click on File menu bar)
4. File pull down menu
5. Click on File/Quit
6. Exit to MS-DOS or FoxPro level
   All CAMD screens are erased before exit.

---

3 CAMD Shutdown
This test assures that the system will not shut down when there is an application window currently open.

1. Start CAMD (see Test 1)
2. Click on Diagnosis/Patient
3. Patient details screen is opened
4. Click on File/Quit
5. Warning pop up: close windows before quitting
6. Close Patient details screen
7. Click on File/Quit
8. Exit to MS-DOS level

---

4 CAMD Menu

This test assures that the CAMD main menu is complete and activates the application modules correctly.

1. Click on Diagnosis
2. CAMD pull down menu is displayed with entries:

    2.1 Patient
    2.2 Encounter
    2.3 Reference

3. Click on Diagnosis/Patient
4. Patient details screen is displayed
5. Click on Cancel or press Esc to remove screen
6. Click on Diagnosis/Encounter
7. Patient Encounter screen is displayed
8. Click on Cancel to remove screen
9. Click on Diagnosis/Reference
10. Medical reference screen is displayed
11. Press Esc to remove screen

---

5 Patients on File Screen

This test assures that the patients on file screen can be called up correctly.

1. Click on Diagnosis/Patient
   (see Test 4.3)
2. Patient details screen is displayed
3. Click on Browse button
4. Patients on file screen is displayed, each entry with:

    4.1 SSN
    4.2 Last name
    4.3 First name

5. Click on an entry
6. Patient details screen (see Step 2) is filled in with information from the selected patient

---

6 Patient Details Screen

This test assures that the patient details screen can be called up correctly.

1. Select a patient
   (see Test 5)
2. Patient details screen with fields:

    2.1 SSN
    2.2 Last name

2.3 First name
2.4 Date of birth
2.5 Sex
2.6 Rate/rank

3. Select a different patient (see Step 1)

---

7 Patient Encounter Screen
This test assures that the patient encounter screen can be called up correctly.

1. Click on Diagnosis/Encounter
(see Test 4.6)
2. Patient encounter screen with fields:

2.1 Date of encounter
2.2 Time of encounter
2.3 Status
2.4 Provider
2.5 Patient Last name
2.6 First name
2.7 SSN
2.8 Age
2.9 Sex
2.10 Complaint
2.11 Diagnosis
2.12 Symptoms
2.13 Assist
2.14 Treatment

---

8 Complaint
This test assures that the complaint is correctly entered.

---

9 Complaint
This test assures that the complaint cannot be altered for an opened encounter.

---

10 Diagnosis
This test assures that the care provider's diagnosis is correctly entered.

---

11 Diagnosis
This test assures that the diagnosis cannot be entered until the complaint field has been completed.

---

12 Symptoms
This test assures that the signs and symptoms are correctly entered.

---

13 Symptoms
This test assures that a numeric data value can be entered.

---

14 Symptoms
This test assures that a multiple check box data value can be entered.

---

15 Symptoms
This test assures that a radio button data value can be entered.

---

16 Symptoms
This test assures that a logical data value can be entered.

---

17 Symptoms
This test assures that the signs and symptoms cannot be entered until the complaint field has been completed.

---

18 Symptoms
This test assures that the signs and symptoms already entered can be edited.

---

19 Assist
This test assures that the computer's differential diagnosis is correctly generated.

---

20 Assist
This test assures that the computer's differential diagnosis will not be generated unless the complaint field has been completed first.

---

21 Assist
This test assures that the computer's differential diagnosis will not be generated unless all significant signs and symptoms have been completed first.

---

22 Treatment
This test assures that the treatment recommendations are correctly entered.

---

23 Treatment
This test assures that the treatment recommendation cannot be entered until the care provider's diagnosis has first been completed.

---

24 Save Encounter
This test assures that the patient encounter is correctly saved to disk.

---

25 Save Encounter
This test assures that an empty patient encounter is not saved to disk.

---

26 Cancel Encounter
This test assures that the cancelled patient encounter is not saved to disk.

---

27 Cancel Encounter
This test assures that the user is prompted to confirm that the patient encounter is being cancelled.

---

28 Print Encounter
This test assures that a patient encounter is correctly printed.

---

29 Print Encounter to File
This test assures that a patient encounter is correctly printed to a disk file.

---

30 Resume Pending Encounter
This test assures that a previous encounter that is pending can be correctly resumed.

Pending encounters screen with fields:

1. SSN (selected patient or blank)
2. Date (today's date)
3. One or more encounters, each with:

3.1 Time of encounter
3.2 Complaint

4. Click on Resume encounter
5. One moment pop up: loading

---

31 Help System
This test assures that the help database is in place and properly activated.

1. Press F1 for help
2. Display help screen with CAMD main menu described
3. Press Esc to remove help screen

4. BATCH TEST PROCEDURE

This section describes the procedures for running and evaluating the batch test. In batch test mode, each test is read sequentially from the driver file, processed, and logged to disk, without any operator intervention. At the end of the test, the summary is displayed, and the program will pause waiting for a key press.

The batch test assumes that the Interactive Test has already been completed.

4.1 Test Command

To run the test procedure, the test driver file and test directory must first have been prepared and loaded onto disk. The test is initiated by running the CAMD expert system with some test parameters. The parameters cause the CAMD expert system to run in batch test mode instead of the normal interactive mode. The test is started with a nMS-DOS command of the following form:

TEST testfile testdirectory testlogfile

where the test parameters are as described next.

4.2 Test Parameters

4.2.1 Test Driver File

testfile is the name of the TEST.DBF driver file. This name is the filename only. The file is assumed to be in the test directory as specified below.

4.2.2 Test Directory

testdirectory is the relative pathname of the TEST TRANSACTION DIRECTORY. All .DBF files required for the test are assumed to be in this directory, and the log file, if written, will be placed in this directory. The pathname must be given relative to the CAMD\DATA directory used by the CAMD expert system.

4.2.3 Test Log File

testlogfile is the name of the test log file. This is an optional parameter, and if it is not given, results are displayed only on the screen.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATE COVERED
	October 1992	Interim - FY92

4. TITLE AND SUBTITLE Naval Shipboard Non-Tactical ADP Program (SNAP) Automated Medical System (SAMS) Computer Assisted Medical Diagnosis (CAMD) Module System/Subsystem Specification

5. FUNDING NUMBERS
Program Element: 63706N
Work Unit Number: 0095.005-6103

6. AUTHOR(S)
Thomas R. Bonifield, Ph.D and James D. Felson, M.D.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Health Research Center
P. O. Box 85122
San Diego, CA 92186-5122

8. PERFORMING ORGANIZATION
Report No. 92-27

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Naval Medical Research and Development Command
National Naval Medical Center
Building 1, Tower 2
Bethesda, MD 20889-5606

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Medical Care Aboard U.S. submarine and small surface ships is rendered by specifically trained independent Duty Corpsmen. To provide these corpsmen with important diagnostic and treatment information during a deployment a Computer Assisted Medical Diagnosis (CAMD) module has been conceived. System specification for this module has been developed to outline the plan for implementing the functions defined in the Functional Description.

14. SUBJECT TERMS
Computer Assisted Diagnosis
Knowledge Base
Expert System
Medical Diagnosis
Bayesian Method
RUle-based System

15. NUMBER OF PAGES 185

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited